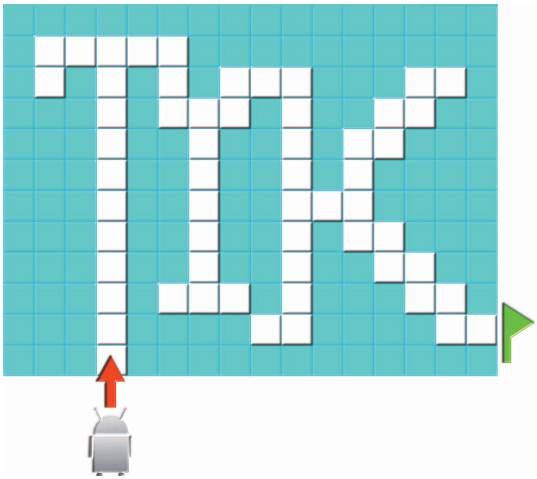
Programowanie i robotyka w edukacji wczesnoszkolnej



Materiały dydaktyczne zawierające treści związane z algorytmiką, wykorzystaniem TIK do sięgania po zasoby elektroniczne i pomoce dostępne w Internecie, zamieszczaniem własnych prac w Internecie, językiem i środowiskiem Scratch oraz robotyką

Kraków 2018

Programowanie i robotyka w edukacji wczesnoszkolnej

Kazimierz Bobko
Magdalena Bubula
Jerzy Marek
Witosław Sala
Paweł Wójciak

Kraków 2018

Programowanie i robotyka w edukacji wczesnoszkolnej

Wydawca:
Małopolskie Centrum Doskonalenia Nauczycieli
ul. Lubelska 23
30-003 Kraków

Autorzy:
Kazimierz Bobko
Magdalena Bubula
Jerzy Marek
Witosław Sala
Paweł Wójciak

Kraków 2018

Materiały dydaktyczne powstały w ramach projektu pt. *Innowacyjne rozwiązania cyfrowe w szkołach podstawowych powiatu nowosądeckiego* dofinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego, w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014-2020, działanie 3.2 Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej, realizowanego od 1 sierpnia 2017 r. do 31 października 2018 r.

ISBN 978-83-88618-10-9

Projekt i opracowanie graficzne, skład, łamanie, druk i oprawa:
Grafpol Agnieszka Blicharz-Krupińska
ul. Czarnieckiego 1, 53-650 Wrocław
tel. 507 096 545, argrafpol@argrafpol.pl

Spis treści

1. Wstęp

Programowanie jako element nauczania realizowanego, zgodnie z nową podstawą programową, począwszy od edukacji wczesnoszkolnej, wymaga od nauczycieli oraz innych osób pracujących z dziećmi i młodzieżą doskonalenia metodyki nauczania tego obszaru wiedzy. Wprowadzone do podstawy programowej nowe treści szczegółowe dotyczące algorytmiki i programowania stały się w dużej mierze nowością również dla nauczycieli. Świadczą o tym m.in. wyniki diagnozy wejściowej zastosowanej w projekcie.

Głównym celem opracowania niniejszego poradnika jest zawarcie wskazówek metodycznych związanych z wprowadzeniem do nauczania, na pierwszym etapie edukacyjnym, elementów algorytmicznego myślenia i programowania oraz robotyki. Ich uzupełnieniem są opracowane w ramach projektu scenariusze zajęć praktycznych, realizowanych z uczniami klas I-III szkoły podstawowej.

Poradnik adresowany jest przede wszystkim do nauczycieli rozpoczynających nauczanie tych zagadnień podczas dodatkowych zajęć praktycznych z dziećmi, a także z myślą o innych osobach dorosłych (bibliotekarzach oraz pracownikach publicznych domów kultury), dla których może on stanowić pomoc dydaktyczną, utrwalającą zakres wiedzy i umiejętności nabytych podczas szkolenia stacjonarnego, realizowanego w ramach projektu pt. „Innowacyjne rozwiązania cyfrowe w szkołach podstawowych powiatu nowosądeckiego".

W związku z tym, iż zajęcia praktyczne były z założenia zajęciami dodatkowymi, a część adresatów poradnika nie realizuje w swej pracy założeń podstawy programowej, autorzy niniejszego opracowania nie koncentrują się na zagadnieniach związanych z formułowaniem wymagań edukacyjnych, ocenianiem czy szerszą podbudową teoretyczną związaną z metodami i technikami kształcenia. Nadrzędnym celem jest rozwijanie kompetencji związanych z technologiami informacyjno-komunikacyjnymi, w tym poprzez:

- rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia;
- programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych, tj.: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Poszczególne części poradnika zawierają wskazówki związane z wykorzystaniem sprzętu i oprogramowania otrzymanego w ramach projektu pt. „Innowacyjne rozwiązania cyfrowe w szkołach podstawowych powiatu nowosądeckiego", w związku z powyższym autorzy pominęli aspekty finansowe oraz zagadnienia bezpiecznego posługiwania się urządzeniami stanowiącymi wyposażenie pracowni komputerowych.

Autorzy zachęcają do nauczania poprzez działanie, doświadczanie i rozwijanie kreatywności.

2. Algorytmika

2.1. Pojęcie algorytmiki

Komputery działając wykonują programy, natomiast programy realizują algorytmy. Programowanie jako proces rozpoczyna się przed uruchomieniem komputera. Aby napisać program komputerowy, nie wystarczy znajomość obsługi środowiska programowania, konieczna jest również wiedza o tym, jak poradzić sobie z problem, dla którego tworzymy rozwiązanie. Niezbędna jest zatem znajomość *algorytmiki*. **Algorytmika to dziedzina wiedzy zajmującą się algorytmami.** Nauczając najmłodszych, powinniśmy od samego początku kształtować tzw. *myślenie algorytmiczne*, czyli rozwijać nawyki myślowe zgodne z zasadami szeroko rozumianej informatyki. Logiczne i algorytmiczne myślenie należy, zgodnie z zapisami nowej podstawy programowej, do najważniejszych umiejętności rozwijanych w ramach kształcenia ogólnego w szkole podstawowej. Zgodnie z nową podstawą programową na pierwszym etapie edukacyjnym uczniowie poznają, często w formie zabawy, nieformalne znaczenie wybranych pojęć związanych z informatyką, takich jak: liniowa kolejność, powtarzanie czynności (sekwencja zdarzeń, logiczny porządek zdarzeń, czynności i wielkości), instrukcja, sekwencje instrukcji (polecenie), algorytm (plan działania). Rozwijanie od najmłodszych lat myślenia i działania algorytmicznego, ma wspomagać stawianie pierwszych kroków w dziedzinie programowania. Wprowadza dzieci w świat języka informatyki, aby stawały się „cyfrowymi twórcami" i odczuwały radość z tworzenia.

Nowa podstawa programowa szczegółowo określa wymagania wobec treści nauczania logicznego i algorytmicznego myślenia w zakresie edukacji informatycznej na I etapie edukacyjnym:

„Osiągnięcia w zakresie rozumienia, analizowania i rozwiązywania problemów. Uczeń:

1) *układa w logicznym porządku: obrazki, teksty, polecenia (instrukcje) składające się m.in. na codzienne czynności;*
2) *tworzy polecenie lub sekwencje poleceń dla określonego planu działania prowadzące do osiągnięcia celu;*
3) *rozwiązuje zadania, zagadki i łamigłówki prowadzące do odkrywania algorytmów"*[1].

Cytowane treści zawarte w podstawie programowej są zapewne przez Państwa realizowane w ramach innych zajęć edukacyjnych (w szczególności punkt 1). Należy jednak pamiętać, aby wzbogacić je o działania mające na celu odkrywanie algorytmów. Według definicji algorytm to **skończony ciąg instrukcji (poleceń) pozwalający wykonać zadanie i rozwiązać problem.**

Wymyślanie algorytmów opiera się niezmiennie na logice. Rozwijając u dzieci logiczne myślenie, kształtujemy kreatywność poprzez rozwiązywanie problemów,

[1] ROZPORZĄDZENIE MINISTRA EDUKACJI NARODOWEJ z dnia 14 lutego 2017 r. w sprawie podstawy programowej … (Dz.U poz. 356)

podejmowanie i realizację innowacji oraz umiejętność optymalizacji działań. Najprościej rzecz ujmując, uczymy je samodzielnie myśleć.

Komputer nie myśli, jest tylko maszyną, która wykonuje polecenia. Krok po kroku musimy mu podawać sekwencję czynności, które ma wykonać, aby rozwiązał dowolny, czasami banalny, problem. Żyjemy i w przyszłości będziemy żyć w skomputeryzowanej cywilizacji wykorzystującej coraz bardziej zaawansowane technologie. Umiejętności programistyczne będą naszym uczniom bardzo przydatne, nawet jeśli nie zostaną zawodowymi programistami.

2.2. Definiowanie problemu i znajdowanie algorytmu rozwiązania, w tym poza środowiskiem programistycznym

Życie człowieka to nie tylko ciągłe rozwiązywanie ważnych dla całej ludzkości problemów ułatwiających istnienie, takich jak wynalezienie koła czy papieru, ale również banalnych rozwiązań związanych z codziennym życiem, np.: jakim środkiem transportu dostać się w określone miejsce, jak przygotować posiłek, czy jak sensownie spakować plecak. Na każdym kroku musimy radzić sobie z problemami, warto więc poznać, wywodzące się z informatyki, sposoby prowadzące do ich rozwiązywania.

Według jednej z definicji, mamy

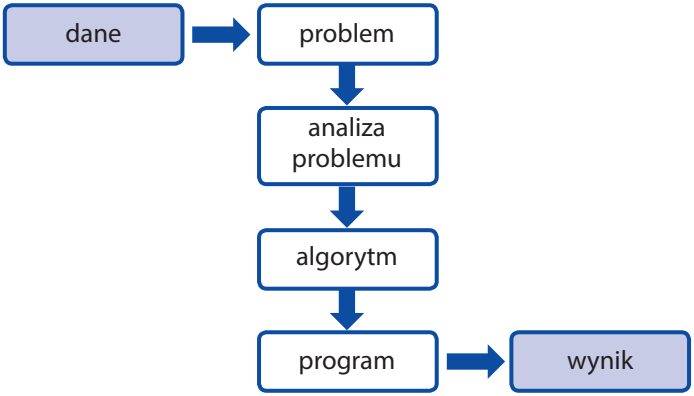
- *„problem, gdy nie podano nam, jak należy go rozwiązać, ale wiemy wystarczająco, by poradzić sobie z nim,*
- *problem jest dla każdego nie tylko dla orłów."*[2]

Tworząc algorytm, powinniśmy określić istotę problemu, czyli dokonać jego analizy, w ramach której należy precyzyjnie określić dane oraz oczekiwany wynik (sformułować tzw. specyfikację, czyli opis elementów tworzących sytuację problemową oraz ewentualne związki miedzy nimi). Następnie, należy przedstawić kolejne kroki rozwiązania problemu – tzn. zbudować algorytm. Niezbędne jest również wykonanie analizy tworzonego rozwiązania pod kątem ewentualnych błędów oraz optymalności. Nie zapomnijmy zatem o przetestowaniu zaproponowanego rozwiązania i sprawdzeniu, czy spełnia nasze oczekiwania. Na podstawie tak przygotowanego algorytmu będzie można stworzyć program w wybranym środowisku programowania. Proces programowania wymaga, aby dzieci uczyły się niezwykle precyzyjnie wyrażać swoje oczekiwania, ponieważ postać na ekranie lub mały robot nie domyślą się, co dziecko chce im przekazać. Dzieci muszą to wyrazić w postaci zrozumiałych poleceń.

Ucząc dzieci podstaw algorytmiki, powinniśmy uwzględniać wszystkie etapy tworzenia rozwiązania danego problemu.

[2] Źródło: Maciej M. Sysło Wprowadzenie do algorytmiki i programowania

Poniżej przedstawiamy schemat czynności prowadzących do rozwiązania problemu z użyciem komputera:



Algorytm różni się od programu bardziej swobodnym zapisem zawierającym opis planu działania. Plan ten można przedstawić w postaci ciągu czynności, które muszą być wykonane w określonej kolejności. Opis czynności (zestaw kroków) występujących w algorytmie nazywamy instrukcjami.

Logiczne myślenie w procesie programowania wymaga poznania i zrozumienia szeregu algorytmów, jak również umiejętności przewidywania konsekwencji danego rozwiązania i planowania działań. Podczas pracy z dziećmi celowe wydaje się zatem, z uwagi na znaczenie algorytmiki w procesie programowania, rozpoczęcie od zajęć odkrywających rozwiązania poza środowiskiem programistycznym – bez użycia komputerów.

2.3. Sposoby przedstawiania algorytmu

Algorytmy powinny być tak przedstawiane, aby było możliwe ich jednoznaczne odczytanie i zastosowanie. W informatyce stosuje się wiele metod przedstawiania algorytmów, m.in.:

- słowny,
- lista kroków,
- schemat blokowy,
- pseudokod,
- język programowania.

Podczas pracy z dziećmi będziemy stosować trzy pierwsze metody.

Opisanie kolejnych czynności w języku potocznym, algorytm słowny, dotyczyć będzie rozwiązania prostych sytuacji, składających się z niewielu elementarnych kroków możliwych do zapamiętania przez uczniów. Na przykład dzieci opowiadają,

w jaki sposób dotarły z domu do szkoły, a różne formy transportu (pieszo, samochodem, autobusem) powodują powstanie odmiennych algorytmów.

Zapisanie opisu słownego w postaci krótkich zdań, mających charakter instrukcji „krok po kroku", nazywane jest listą kroków. Można więc w pewnym sensie uznać, że książka kucharska, jako zestaw przepisów kulinarnych, opisujących krok po kroku receptury przygotowania potraw, jest przykładem zbioru algorytmów. Jednakże przepisy nie zawsze są „precyzyjne", a takie są wymagania w stosunku do algorytmów. Przedstawienie algorytmu w postaci listy kroków jest proste, ale w przypadku bardziej złożonego problemu, większej ilości kroków, staje się mało czytelne.

Przykład zapisu w postaci listy kroków mających doprowadzić do rozwiązania problemu polegającego na przeniesieniu plecaka z jednego miejsca na drugie:

Krok 1. Idź prosto 5 kroków.

Krok 2. Obróć się w prawo.

Krok 3. Idź prosto 3 kroki.

Krok 4. Podnieś plecak.

Krok 5. Obróć się 2 razy w lewo.

Krok 6. Idź prosto 7 kroków.

Krok 7. Obróć się w lewo.

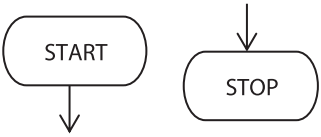
Krok 8. Idź prosto 2 kroki.

Krok 9. Połóż plecak.

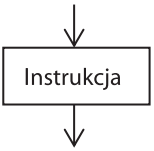
Tworząc polecenie rozwiązania problemu, instrukcje do wykonania, stosujemy zwykle pewien rodzaj uproszeń. Możemy pominąć pewne elementarne czynności i, tak jak w powyższym przykładzie, nie musimy opisywać dokładnie instrukcji wykonania polecenia „idź" (podnieś jedną nogę - przesuń do przodu – postaw - podnieś drugą nogę – dostaw, itp.) czy „połóż plecak". Podobnie odbywa się to w praktyce – środowiska programistyczne mają bowiem przygotowane zestawy instrukcji dotyczące czynności elementarnych.

Kolejnym sposobem prezentowania algorytmu jest zapis graficzny przy użyciu tzw. schematu blokowego. Aby zapisać algorytm za pomocą takiego schematu, należy poznać stosowane symbole i ich znaczenie. Rozpoczynając naukę algorytmiki, do tworzenia umownego kodu będziemy niejednokrotnie stosować własne symbole, takie jak strzałki, odcisk stopy, itp. Formalny zapis obowiązujący w informatyce to zestaw bloków o określonych kształtach łączonych za pomocą strzałek.

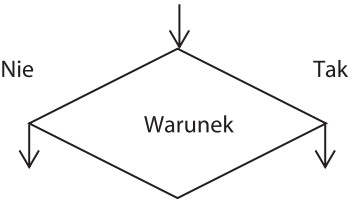
W ten sposób pokazujemy kolejność wykonywania instrukcji.



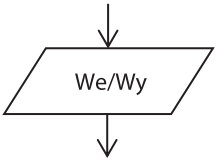
Początek i koniec każdego algorytmu (bloki graniczne)



W bloku instrukcji umieszcza się polecenia do wykonania (instrukcje) - obliczenia, wprowadzenie wartości



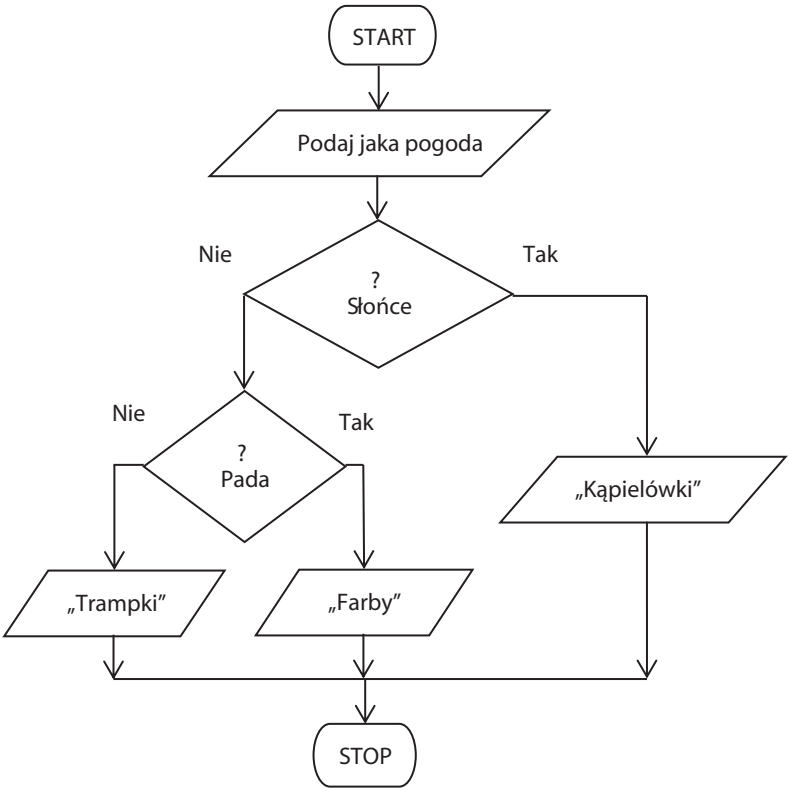
Blok warunkowy, który decyduje o wyborze dalszej drogi postępowania



W bloku wejścia/wyjścia umieszcza się wprowadzane dane lub wyprowadzane wyniki

Przykład użycia schematu blokowego do rozwiązania problemu „Poprawne spakowanie tornistra".

Pani nauczycielka prosi dzieci, aby następnego dnia przyniosły do szkoły kąpielówki, gdy będzie świecić słońce, trampki, jeśli niebo będzie zachmurzone, lub, w przypadku deszczu, farby do malowania. Janek, żeby się nie pomylić, użyje do pakowania plecaka robota, którego musi zaprogramować.



Schemat blokowy, stanowiący swoistą sieć działań, jest jedną z najbardziej popularnych form przedstawiania algorytmu. Jego podstawową zaletą jest łatwość analizy.

Ucząc dzieci w klasach I-III, można posłużyć się uproszeniem schematów blokowych stosując grafy używane w nauczaniu wczesnoszkolnym .

Algorytmy mogą być:

Proste (sekwencyjne) - nie używa się w nich bloków warunkowych. W takiej sieci działań kolejność realizacji poszczególnych operacji jest ściśle ustalona, przy czym żadna z nich nie może być pominięta ani powtórzona.

Z rozwidleniem (warunkowe) - zawiera w sobie wybór jednej z kilku możliwych dróg realizacji danego zadania, tj. przynajmniej jeden blok warunkowy.

Z pętlą (iteracyjne) – gdy w trakcie realizacji danego zadania konieczne jest powtórzenie niektórych operacji różniących się jedynie zestawem danych. Pętla obejmuje tę część bloków, która ma być powtórzona.

W praktyce algorytmy występują najczęściej jako kombinacje powyższych rodzajów. Pozostałe sposoby przedstawiania algorytmów (pseudokod, język programowania) wymagają znajomości składni języków programowania, stąd nie są zalecane na wczesnych etapach kształcenia.

2.4. Metody analizy oraz porównywanie dostępnych możliwości rozwiązania problemu/sytuacji problemowej - wybór najlepszej metody

W przypadku dowolnego problemu chcemy otrzymać rozwiązanie komputerowe, które jest przede wszystkim poprawne, tzn. spełnia specyfikację (opis) problemu, oraz na tyle efektywne, że nie marnuje czasu i pamięci komputera.

Tworzone algorytmy powinna charakteryzować:

- **poprawność** – prawidłowe „działanie" prowadzące do osiągnięcia rozwiązania problemu (oczekiwanych wyników);
- **jednoznaczność** – zawsze te same wyniki działania przy tych samych danych wejściowych;
- **szczegółowość** – poziom szczegółowości zapisu uzależniony od stopnia skomplikowania problemu oraz warunków wykonania;
- **skończoność** – ograniczone (czasowo) wykonywanie kolejnych poleceń, tj. skończona liczba kroków;
- **sprawność** – szybkość działania, złożoność czasowa określana również efektywnością algorytmu.

Wymienione cechy algorytmu należy brać pod uwagę podczas analizy prezentowanych algorytmów pamiętając, że nawet uzyskanie poprawnego pod względem działania rozwiązania problemu nie zwalnia nas z poszukiwania innych rozwiązań, które mogą okazać się bardziej efektywne. W przypadku nauczania dzieci, wystarczające wydaje się określenie liczby elementarnych kroków w prezentowanym/ch rozwiązaniu/ach. Ważne jest, aby analizy algorytmów, jako elementu rozwijania logicznego myślenia, nie pomijać podczas pracy z dziećmi i młodzieżą.

Proponowane rozwiązania problemu/sytuacji problemowej sprawdzamy (warto to wykonywać wspólnie z dziećmi) pod względem poprawności, czyli poprzez prześledzenie „krok po kroku" kolejnych poleceń, podając przykładowe dane. **W ten sposób symulujemy wykonanie algorytmu**.

Kluczową umiejętnością podczas nauki programowania jest analizowanie złożonych czynności i rozkładania ich na czynności proste. Problem należy dokładnie wyjaśnić i opisać, a następnie wspólnie przeanalizować. Odkrywanie i prezentacja wszystkich możliwych rozwiązań, które nasuwają się uczniom, pomaga w rozwijaniu algorytmicznego myślenia. Po wyczerpaniu się wszystkich pomysłów grupa wybiera najbardziej odpowiednie – optymalne – rozwiązanie, stosując np. metodę burzy mózgów.

2.5. Ćwiczenie umiejętności przekładania prostych działań związanych z życiem codziennym na algorytm

Rozwijając umiejętność logicznego myślenia, tj. przedstawiania prostych działań w postaci algorytmu, nauczyciel powinien wykorzystywać treści związane z wszystkimi obszarami edukacji. Należy zacząć od najprostszych problemów, np.:

- jak poprawnie zgłosić się do odpowiedzi:

podnieś rękę – czekaj, aż nauczyciel zezwoli Ci na udzielenie odpowiedzi – wstań – udziel odpowiedzi – kiedy skończysz, usiądź;

- jak przejść przez jezdnię z sygnalizacją świetlną:

zatrzymaj się przed przejściem dla pieszych - jeśli świeci się czerwone światło, stój – jeśli świeci się zielone światło, upewnij się, czy nie jedzie żaden pojazd, a następnie przejdź przez jezdnię.

Stopniowe wprowadzanie słów związanych z myśleniem algorytmicznym umożliwia dzieciom poznanie, jak „rozumuje komputer". Warto zachęcać uczniów, aby w konkretnych sytuacjach prezentowali własne rozwiązania układając sekwencje zdarzeń w logicznym porządku. W ten sposób, poznają intuicyjnie takie pojęcia, jak pojedyncza instrukcja, sekwencja warunku oraz wiele innych. Pozwalając im formułować polecenia, czyli opisywać kolejne kroki dla wybranego obiektu nawet w prostych działaniach, rozwijamy u dzieci wyobraźnię. Często w ramach innych zajęć edukacjnych uczniowie proszeni są o ułożenie rozsypanych obrazków w określonym porządku. Tego rodzaju ćwiczenia, które mogą być inspiracją do tworzenia planów działania – algorytmów, m.in. obejmują: układanie obrazków w takiej kolejności, żeby zrobić kanapkę, zaparzyć filiżankę herbaty, przedstawić hodowlę fasolki, itp.

Jeśli uczniowie potrafią czytać, obrazki mogą być zastąpione tekstem, co pozwoli na układanie bardziej złożonych sekwencji czynności.

Wprowadzając dzieci w świat programowania, nauczyciel przede wszystkim tworzy warunki, dostosowując zakres metod nauczania do określonej grupy uczniów, aby umożliwić każdemu z nich przyswojenie zagadnień.

2.6. Kształtowanie postawy twórczej poprzez wymyślanie prostych zadań algorytmicznych

Tytuł ten należy rozumieć jako informatyczne podejście do rozwiązywania problemów z różnych dziedzin. Umiejętność ta jest systematycznie rozwijana poprzez wykorzystanie zabaw i gier interakcyjnych czy gier planszowych w połączeniu z różnymi formami aktywności poznawczej ucznia w młodszym wieku szkolnym. Dziecko uczy się dzięki samodzielnemu poznawaniu w trakcie działania, a nieocenioną funkcję w procesie nauki odgrywa zabawa. Zachęcamy uczniów do tworzenia mapy skojarzeń składającej się z obrazków i symboli połączonych różnymi gałęziami czy strzałkami.

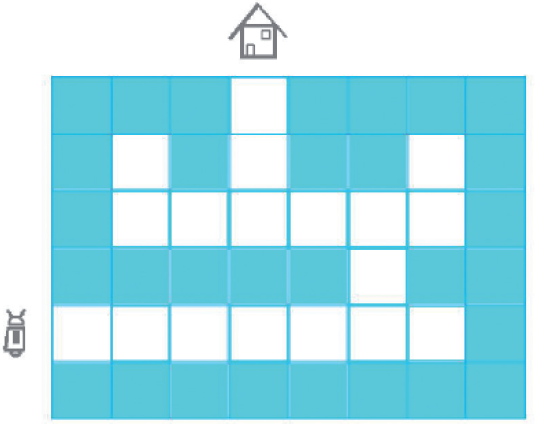
Przykład zadania dla uczniów: Pokieruj robotem tak, żeby dotarł do domku.

Zadaniem uczniów jest ułożenie algorytmu składającego się z symboli, za pomocą których kodowane są elementarne instrukcje wydawane obiektowi w postaci robota:

Idź (krok)

Obróć się w lewo

Obróć się w prawo

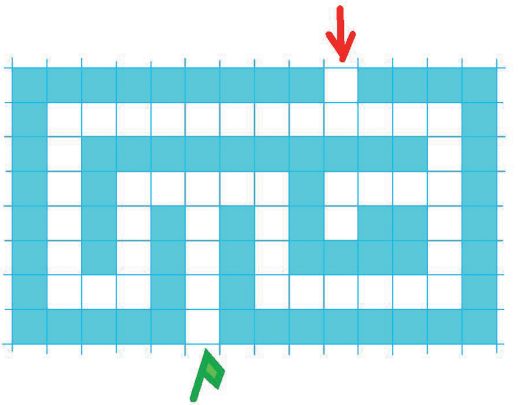


Uczniowie tworzą kod używając symboli:

Można zaproponować również zapis skrócony w postaci, która scala powtarzające się instrukcje:

6x  1x  2x  1x  2x  1x  3x 

Przykład ćwiczenia: Zaproponuj najkrótszą drogę przejścia przez labirynt (od startu do mety).



W podanym przykładzie uczniowie wyznaczają prawidłowe rozwiązanie, zliczając elementarne kroki od startu do mety. Szereg podobnych ćwiczeń polegających na poruszaniu się w labiryncie można wykonać wykorzystując plansze, maty edukacyjne, a nawet płytki na podłodze w pomieszczeniach szkoły.

Poprzez stwarzanie sytuacji skłaniających uczniów do logicznego, algorytmicznego myślenia w ramach różnych form edukacji, nauczyciel realizuje cały szereg wymagań dotyczących „rozumienia, analizowania i rozwiązywania problemów". Przygotowuje tym samym swoich uczniów do późniejszego pisania programów.

3. Praca w chmurze

3.1. Wykorzystanie TIK do sięgania po zasoby elektroniczne i pomoce dostępne w Internecie

Praca w chmurze to innowacyjna technologia wykorzystująca komputer osobisty jedynie jako rodzaj wtyczki, tj. narzędzie niezbędne do podłączenia się do sieci, gdzie dostęp do danych staje się możliwy po zalogowaniu się na wybranej stronie. Wiele aplikacji działających w chmurze nie wymaga instalacji żadnego oprogramowania, ponieważ korzystają z przeglądarki internetowej. Wersje na smartfony używają specjalnych aplikacji, ale z uwagi na mały ekran nie są wygodne w obsłudze. W Internecie można znaleźć wiele stron i aplikacji do wykorzystania na pierwszym etapie edukacyjnym. Nauczyciel powinien sam zdecydować, jakie strony i aplikacje będą odpowiednie dla jego uczniów.

Kolejny rozdział podaje przykładowe strony do wykorzystania.

3.1.1. Strony WWW z aplikacjami działającymi on-line

Kodable https://www.kodable.com/ - program uczący programowania adresowany do najmłodszych. Zadaniem ucznia jest napisać właściwy układ drogi za pomocą strzałek. Fuzzi musi przemieścić się od punktu początkowego do końca trasy, zbierając po drodze monety.

Scratch https://scratch.mit.edu/ - prosta aplikacja do kodowania wizualnego, polegająca na składaniu algorytmu z kolorowych bloków. Można z jej pomocą tworzyć proste gry i animacje.

Code.org https://code.org/ - zestaw kursów przeznaczonych dla każdego przedziału wiekowego (**4-18 lat).**

Kodologia https://kodologia.pl/ - strona udostępniająca 20 kursów dotyczących programowania.

CodeMonkey https://www.playcodemonkey.com/# - prosta aplikacja dla najmłodszych.

Run Marco https://www.allcancode.com/runmarco - aplikacja w języku polskim, w której piszemy polecania za pomocą prostego kodu.

Code Combat https://codecombat.com/ - platforma dla uczniów, przeznaczona do nauki programowania w języku Python i JavaScript, na której przechodzą przez kolejne poziomy gry za pomocą konkretnych komend.

Blockly https://blockly-games.appspot.com/ - uczy podstawowych zasad programowania i daje dzieciom wprowadzenie do języka JavaScript.

Tynker https://www.tynker.com/ - dzieci uczą się tworzyć własne gry i aplikacje, a także programować mody w świecie Minecraft.

Code Avengers https://www.codeavengers.com/ - strona oferuje kursy wprowadzające do programowania, tworzenia stron internetowych i kodowanie w języku Python oraz JavaScript. Dostęp do platformy jest darmowy przez limitowany okres czasu.

Code Monster http://www.crunchzilla.com/code-monster - interaktywna gra, która uczy dzieci pisania skryptów w języku JavaScript.

3.1.2. Strony WWW z aplikacjami do pobrania na Androida

Scottie go! - połączenie gry planszowej oraz aplikacji, nauka programowania z przyjaznym ludzikiem Scottiem. Zadaniem ucznia jest zaprogramować ruchy ludzika.

https://play.google.com/store/apps/details?id=com.netictech.scottiegoedu&hl=pl

ScratchJr - wprowadza język programowania, przeznaczona dla dzieci od 5 roku życia, pozwala na tworzenie animacji oraz prostych gier. Dzieci łączą graficzne bloki programistyczne, aby postaci poruszały się, skakały, tańczyły czy śpiewały.

Photon – interaktywny robot, którego dzieci mogą rozbudowywać, pokonując kolejne poziomy.

The Foos - gra na smartfony i tablety z Androidem, która nie tylko dostarcza mnóstwo frajdy, ale też pozwala zrozumieć zasady programowania.

Minecraft - https://code.org/minecraft połączenie gry z językiem Python, umożliwiającej dzieciom naukę programowania i matematyki.

Istnieje wiele witryn pomagających dzieciom w nauce programowania, wymieniliśmy jedynie te najbardziej popularne.

3.2. Opis aplikacji Kodable oraz code.org

3.2.1. Kodable

Aplikacja uczy podstaw kodowania i dedykowana jest dzieciom już od 5 roku życia. Za pomocą poleceń programujemy trasę włochatego stwora o imieniu Fuzzi. Jego zadaniem jest zebranie wszystkich monet znajdujących się w labiryncie. Oprócz prostych poleceń związanych z kierunkiem ruchu, na wyższych poziomach stosujemy również warunki i powtórzenia. Instrukcje wizualne oraz poziomy

pozwalają dzieciom uczyć się krok po kroku koncepcji programowania, zanim jeszcze nauczą się czytać.

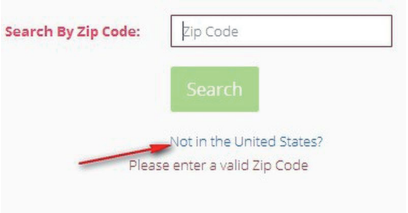
Zaletą tej aplikacji jest to, że uczeń musi przejść wszystkie ćwiczenia po kolei. Jeżeli popełni błąd, aplikacja podpowiada, w którym miejscu się to zdarzyło.

Logowanie się do serwisu

Wchodzimy na stronę https://www.kodable.com/

Zakładamy konto

Po wypełnieniu formularza rejestracyjnego przechodzimy dalej. Należy zwrócić uwagę na komunikat *„Not in the United States?"* i wybrać go klikając myszką, w przeciwnym razie nie przejdziemy dalej.



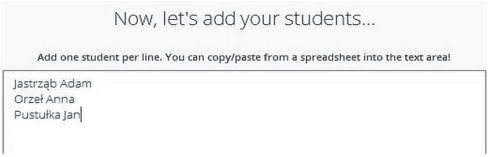
Zakładanie klasy

Będąc już na stronie głównej Kodabli klikamy na przycisk →

W polu *Class Name* wpisujemy nazwę swojej klasy. W polu *Assign a grade* wybieramy 1st Grade (pierwszy poziom lub wyższy w zależności od umiejętności klasy) .

Dodawanie uczniów

Na jednej stronie możemy wpisać wszystkich uczniów (rys. obok) lub dodawać ich pojedynczo. Po wpisaniu uczniów zatwierdzamy wykonane działanie *(Save Students and Continue)*.

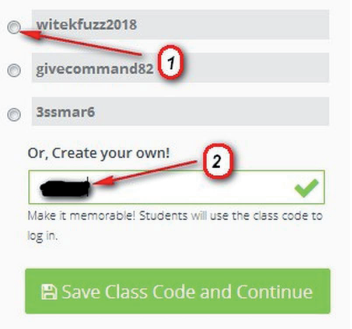


Na następnej stronie *(Select your class code)* nadajemy kod klasie.

Kod klasy będzie potrzebny uczniom do zalogowania.

Mamy tu dwie opcje:

- wybrać kod proponowany przez aplikację;
- wpisać własny.



Zapisujemy i przechodzimy do swojej klasy *(Go to My Class)*

Nadawanie uczniom loginów i haseł

Aby nadać uczniom loginy i hasła:

1. Wybieramy z menu **„Students"**.
2. Wybieramy funkcję koloru niebieskiego (**Print Parent Instructions All Students -** rys. poniżej).
3. Wybieramy nowe okno **„Print Parent Instructions"** (kolor zielony).

4. Drukujemy kody dla ucznia.

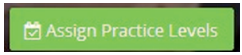
Dodawanie ćwiczeń

Przypisywanie poziomów do poszczególnych lekcji wygląda następująco.

Aby przypisać pierwszy poziom lekcji, należy wykonać poniższe działania:

1. Wybrać klasę z pulpitu nawigacyjnego.
2. Wybrać wyróżnioną lekcję (*jest to zależne od tempa zajęć*).
3. Kliknąć pole z nazwą wybranej lekcji (*np. Algorithms*).

Po wykonaniu tych poleceń otworzy się nowe okno z opisem wybranej lekcji. W prawym górnym rogu opisu lekcji należy wybrać pole z napisem **Assign Practicle Levels** (*Przypisywanie poziomów praktycznych)*



4. Kliknąć „OK".

Dzięki temu 

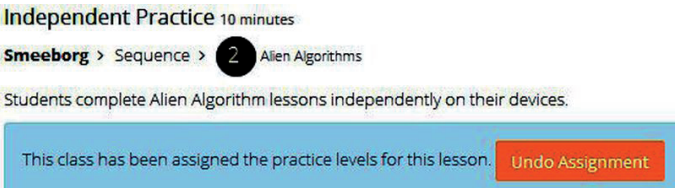
uczniowie będą mieli dostęp do poziomu lekcji.

Zielony znacznik wyboru obok lekcji oznacza, że poziom danej lekcji został przypisany uczniom.

Anulowanie poziomów lekcji

Aby ponownie zablokować (cofnąć) wybrany poziom lekcji, należy:

1. Wybrać klasę z pulpitu nawigacyjnego.
2. Wybrać lekcję, którą chcemy zablokować.
3. Po wejściu do opisu lekcji przewinąć ją do momentu, aż znajdziemy pole z napisem „**Undo Assignment**" (*cofnij przypisanie*) (rys. poniżej). Klikamy na to pole, żeby zatwierdzić polecenie.



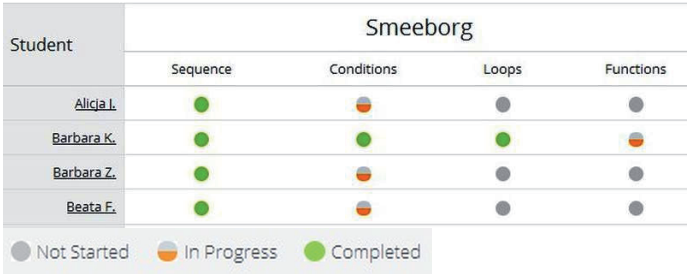
Administrowanie klasą

Nauczyciel może dokonywać wielu zmian:

- dodawać nowych uczniów lub ich usuwać;
- dodawać nową klasę;
- przypisać poziom;
- dodać dostęp do nowych ćwiczeń lub go zablokować;
- sprawdzać postępy uczniów oraz całej klasy.

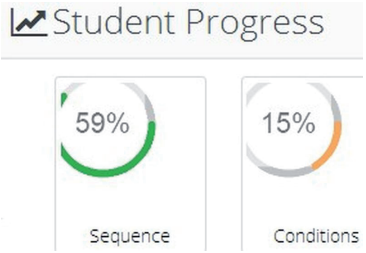
Sprawdzanie postępów klasy i ucznia

Postępy klasy obrazuje poniższa tabelka. Zielony kolor pokazuje wykonanie wszystkich ćwiczeń z danego działu, szary kolor informuje, które zadania nie zostały jeszcze rozpoczęte, kolor pomarańczowy informuje, że uczeń zrobił je częściowo.



Postępy ucznia

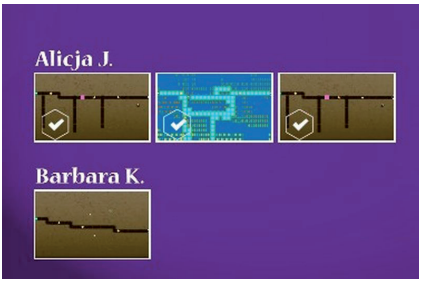
Dzięki przedstawionym wykresom otrzymujemy szczegółową informację o procencie wykonanych zadań w danym zakresie przez danego ucznia.



Umieszczanie własnych prac

Po uruchomieniu programu możemy sami zaprojektować grę. W tym celu należy kliknąć na przycisk „**Create**".

Po wykonaniu możemy ją zapisać.



Wszystkie prace klasy znajdą się w zakładce Your Class.

3.2.2. Code.org

Strona znajduje się pod adresem: https://code.org

Celem aplikacji jest przybliżenie i wprowadzenie dzieci w świat programowania i myślenia algorytmicznego poprzez zabawę. Każdy etap poprzedzony jest nagraniem wideo, które informuje, co należy zrobić.

Na stronie głównej Code Studio, w zakładce Katalog Kursów, znajdują się następujące kursy z Podstaw Informatyki:

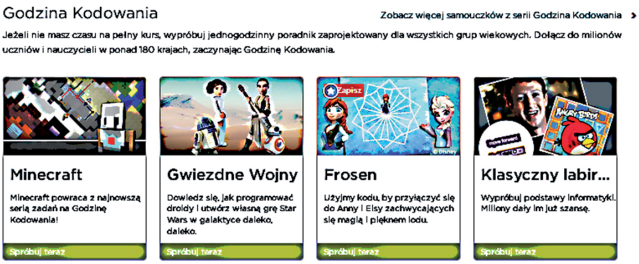
Kurs 1 – dla początkujących w wieku 4-6 lat;

Kurs 2 – dla uczniów powyżej 6 roku życia, którzy potrafią czytać;

Kurs 3 – dla uczniów w wieku 8-18 lat - kontynuacja Kursu 2;

Kurs 4 – dla uczniów, którzy powinni wcześniej ukończyć Kursy 2 i 3, w wieku 10-18.

Ponadto na wymienionej stronie można znaleźć inne kursy przedstawione poniżej.

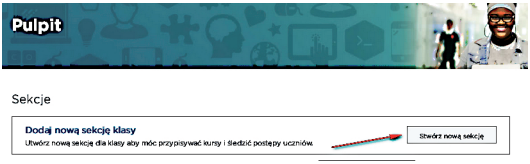


Logowanie w serwisie

Po wejściu na stronę https://code.org należy wypełnić formularz rejestracyjny.

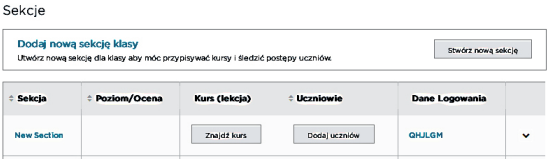
Zakładanie klasy

Po zarejestrowaniu wchodzimy na pulpit i tworzymy nową klasę, klikając na **„Utwórz klasę”.**



Dodawanie uczniów

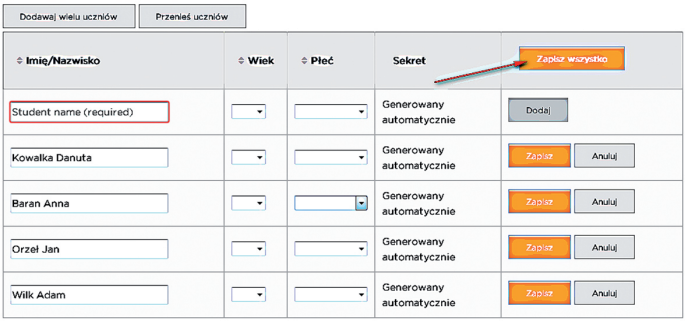
Uczniów możemy dodawać pojedynczo lub grupowo.



Nadawanie uczniom loginów i haseł

W nowej sekcji wybieramy sposób, w jaki będą się logować uczniowie. Mamy do wyboru loginy obrazkowe lub słowne.

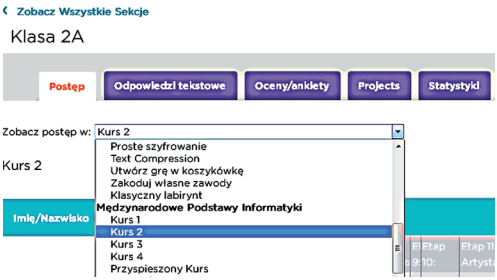
Po dodaniu wszystkich uczniów klikamy **„Zapisz wszystko”**(*rys. poniżej*).



Poniżej karty z nazwiskami uczniów wybieramy pole „Wydrukuj karty z Informacjami o logowaniu dla swoich uczniów” i drukujemy karty do logowania. Uczniowie znajdą na karcie wszelkie niezbędne informacje, aby zalogować się na kursie.

Dodawanie kursów:

- wejdź na pulpit;
- wybierz klasę, której chcesz przypisać kurs;
- wybierz kurs (rys. poniżej).



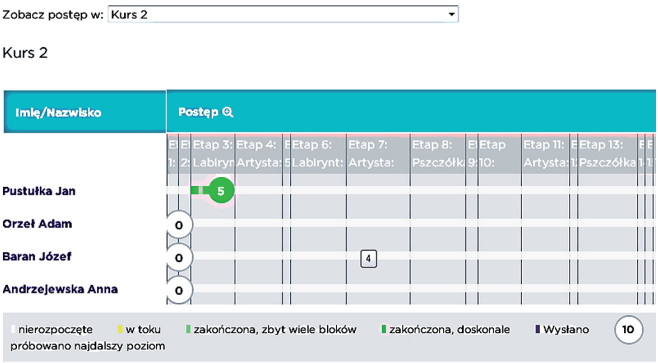
Administrowanie klasą

Nauczyciel może dokonywać następujących zmian:

- dodawać nowych uczniów lub ich usuwać;
- dodawać nową klasę;
- przypisać kurs;
- sprawdzać postępy uczniów oraz całej klasy.

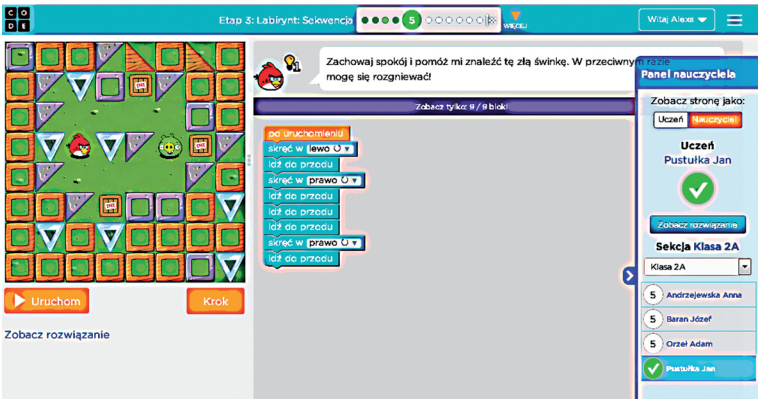
Sprawdzanie postępów

Z pozycji pulpitu klikamy na klasę, której postępy chcemy sprawdzić. Aby zobaczyć więcej informacji, np. które zadania uczeń rozwiązał, należy kliknąć na lupkę.



Nauczyciel może przeanalizować sposób rozwiązania zadań przez konkretnego ucznia. W tym celu powinien:

- kliknąć na Kurs z pozycji pulpitu;
- wybrać klasę, następnie wybrać ucznia;



- wybrać zadanie.

Umieszczanie własnych prac

Każdy uczeń może stworzyć własną pracę. W tym celu powinien wejść na stronę główną w menu **„Galeria Projektów”** i wybrać kategorię, której chce użyć do zrobienia pracy (np. Play Lab, Artysta, App Lap). Uczeń może pracować nad projektem etapami, a po zakończeniu udostępnić wszystkim efekty swojej pracy.

3.3. Konkurs Informatyczny Bóbr

Informacje o konkursie znajdują się na stronie: https://www.bobr.edu.pl/o-konkursie/.

„Bóbr to polska nazwa powołanego do życia w 2004 roku na Litwie międzynarodowego konkursu Bebras z zakresu informatyki oraz technologii informacyjnej i komunikacyjnej. Konkurs Bóbr jest adresowany do uczniów we wszystkich typach szkół. Udział w Konkursie nie jest związany z żadna opłatą, w przyszłości jednak nie wyklucza się wprowadzenia opłat na wzór innych konkursów, takich jak Kangur.

Cele konkursu

Głównym celem konkursu jest rozwój i kształtowanie myślenia algorytmicznego i komputacyjnego oraz popularyzacja posługiwania się technologią informacyjno-komunikacyjną wśród wszystkich uczniów na wszystkich etapach edukacyjnych.

W szczególności konkurs ma na celu:

- zwiększenie zaangażowania uczniów w stosowaniu komputerów i technologii informacyjnej podczas poznawania różnych dziedzin

od samego początku pobytu w szkole, przyczynianie się do rozwoju i kształtowania twórczego podejścia przy zdobywaniu wiedzy i umiejętności,

- sprzyjanie wyrównywaniu szans stosowania tej technologii w grupach dziewcząt i chłopców posiadających dostęp do komputerów, jak i nie mających takiego dostępu,
- zachęcanie uczniów do zdobywania umiejętności potrzebnych w ich życiu osobistym i w przyszłej pracy zawodowej,
- stwarzanie okazji do wymiany doświadczeń między uczniami i nauczycielami z różnych szkół w różnych krajach.

Zadania konkursowe

Charakter zadań konkursowych ma duży wpływ na zainteresowanie konkursem. Zadania w konkursie Bóbr są na ogół związane z posługiwaniem się komputerem i jego oprogramowaniem przy rozwiązywaniu różnych sytuacji i problemów, pochodzących z różnych zastosowań i przedmiotów nauczania, takich jak: język ojczysty, język obcy, geografia, historia, sztuka, technika i matematyka. Niektóre zadania dotyczą budowy komputerów, inne oprogramowania użytkowego. Zadania często są związane również z kulturą i językiem. Wiele zadań umożliwia uczniom wykazanie się umiejętnościami algorytmicznego myślenia.

Zadania są w formie testów. Aby je rozwiązać należy: wybrać jedną poprawną odpowiedź spośród na ogół czterech możliwych, lub wpisać odpowiedź w okienku, lub wykonać inne czynności prowadzące do rozwiązania, o których jest mowa w zadaniu. Rozwiązania można poprawiać."[3]

Kategorie konkursu

Skrzat (klasy I-III szkoła podstawowa)

Benjamin (klasy IV-VI szkoła podstawowa)

Junior (gimnazjum)

Senior (szkoły ponadgimnazjalne)

O zadaniach

Konkurs Bóbr, niezależnie od kategorii, składa się z 24 pytań podzielonych na trzy grupy pod względem stopnia trudności, w każdej grupie znajduje się 8 pytań.

Jak wykorzystać zasoby z konkursu Bóbr

Po wejściu na stronę https://www.bobr.edu.pl/ klikamy w zakładkę „**Zadania i wyniki."** Tam znajdziemy zadania z lat 2016-2017.

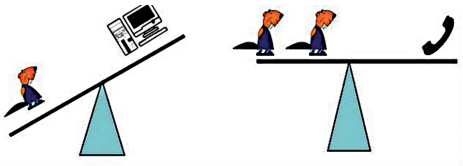
Więcej zadań archiwalnych z lat 2011-2017 dostępnych jest na darmowej platformie e-learningowej mCourser (https://www.mcourser.pl) w wersji interaktywnej. Aby uzyskać do nich dostęp, wystarczy się zarejestrować i zalogować.

[3] Źródło: https://www.bobr.edu.pl/o-konkursie/

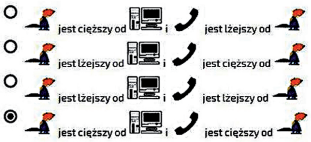
Skrzat (klasy I-III Szkoła podstawowa)

Wersja konkursowa

Wersja z odpowiedziami



Które z poniższych stwierdzeń jest prawdziwe?



Przykład zadania

Zadania można rozwiązywać z dziećmi i zobaczyć naszą punktację.

Wyniki:	Błędy	Punkty
Dostawca	0	0
Powiększanie pokoju	0	0
Poszukiwanie skarbu	0	0
Przyciski typu radio	0	3/3
Bobry na szali	0	3/3
Siatka w RGB	0	0
Miasta na Litwie	0	0
Czarno-białe obrazy	0	0
Czy Bóbr złamał prawo?	0	4/4

Przykład punktacji

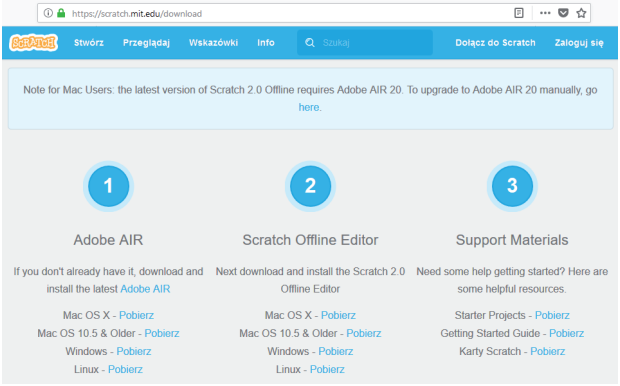
4. Środowisko Scratch

4.1. Pobieranie i uruchamianie programu w środowisku Scratch - zapis pracy

Zgodnie z wprowadzoną w 2017 r. nową podstawą programową kształcenia ogólnego umiejętność programowania stała się jednym z elementów kształcenia powszechnego. Programowanie definiowane jest jako proces rozpoczynający się od specyfikacji problemu, a kończący się testowaniem opracowanego rozwiązania za pomocą odpowiednio dobranej aplikacji lub języka programowania. Tak rozumianego programowania mają być nauczane dzieci od najmłodszych lat, aby wykształcić u nich logiczne myślenie oraz precyzyjne prezentowanie myśli i pomysłów. Poprzez programowanie uczniowie będą mogli również nauczyć się dobrej organizacji pracy oraz rozwijać kompetencje potrzebne do pracy w zespole i efektywnej realizacji projektów.

W ramach edukacji informatycznej na etapie wczesnoszkolnym uczniowie programują wizualnie proste sytuacje lub historyjki, pojedyncze polecenia oraz ich sekwencje sterujące obiektem na ekranie komputera lub innego urządzenia cyfrowego. [4]

Aby wprowadzić młodych uczniów w świat programowania wizualnego, możemy sięgnąć po darmowe środowisko edukacyjne Scratch stworzone przez Massachusetts Institute of Technology. Środowisko przeznaczone jest dla uczniów w wieku 8–16 lat i umożliwia tworzenie historyjek interaktywnych, gier i animacji. Można je pobrać na stronie: https://scratch.mit.edu/download.

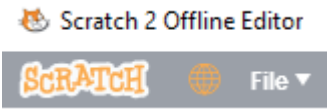


Krok 1 - pobieramy i instalujemy aktualną wersję Adobe AIR. Krok 2 - pobieramy i instalujemy środowisko Scratch Offline Editor. Krok 3 - otrzymujemy dostęp do materiałów, w języku angielskim, wprowadzających nas w programowanie w Scratchu. Dodatkowe materiały, również w języku angielskim, znajdziemy na stronie https://scratch.mit.edu/tips. Przedstawione tam przykłady projektów mogą

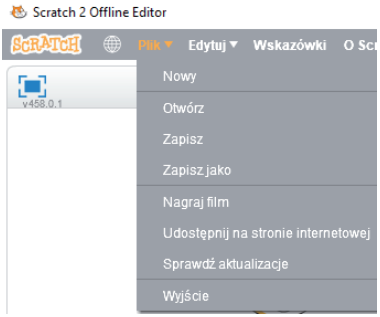
[4] Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła podstawowa. Informatyka, dostępna online: https://www.ore.edu.pl/wp-content/uploads/ 2017/05/ informatyka.-pp-z-komentarzem.-szkola-podstawowa-1.pdf (dostęp dn. 17.02.2018 r.)

służyć jako inspiracja również tym z Państwa, którzy nie znają języka angielskiego, ale zapoznają się z treścią niniejszego rozdziału.

Aby pracować w zainstalowanym przez nas edytorze zmieniamy język na polski, korzystając z ikony globusa:

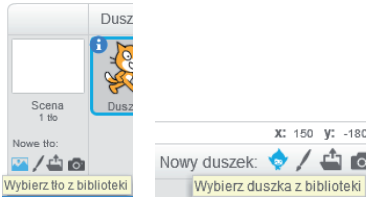


Korzystając z Menu -> Plik możemy stworzyć nowy program komputerowy, zapisać wynik naszej pracy oraz otworzyć zapisany program w pliku znajdującym się na dysku twardym naszego komputera lub pamięci zewnętrznej typu pendrive/ memory stick. Stworzony przez nas program możemy również udostępnić na stronie https://scratch.mit.edu/ po uprzednim utworzeniu własnego konta.



4.2. Poznanie środowiska Scratch

Po uruchomieniu środowiska widzimy duszka – kota znajdującego się na białym tle. Biblioteka Scratcha pozwala wybrać inne tło lub duszka.



Możemy również namalować własne tło lub duszka czy użyć tła albo duszka zapisanego w pliku lub nagraniu, które sami stworzymy. W tym rozdziale zapoznamy się z propozycjami wykorzystania tła i duszków dostępnych w bibliotece oraz z wybranymi możliwościami zakładek: skrypty, kostiumy i dźwięki.

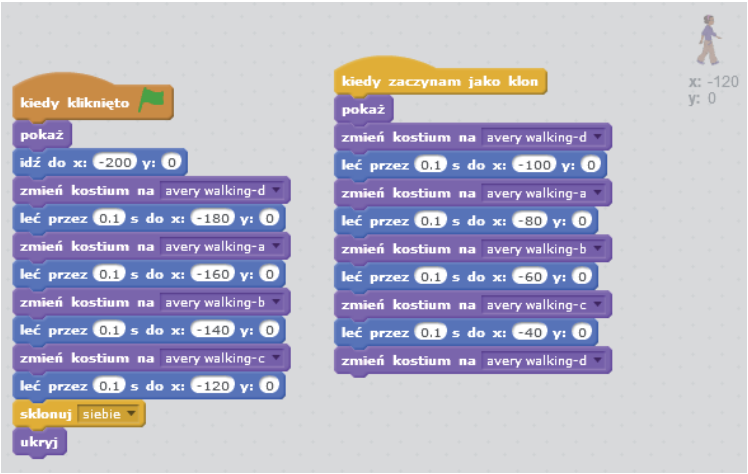
Ćwiczenie 1: Pierwsze kroki.

Wybieramy tło „Berkeley mural" oraz duszka „Avery walking-a". Aby duszek mógł poruszać się na scenie, należy stworzyćprosty skrypt:

Program uruchomi się po kliknięciu zielonej flagi u góry sceny. Zobaczymy duszka przesuwającego się w prawo. Bloczki na granatowym tle odpowiadają za ruch. Proponujemy przećwiczyć inne opcje: „przesuń", „obróć", „ustaw kierunek", zmieniając jednocześnie edytowalne wartości na białym tle bloczków.

Ćwiczenie 2: Animacja kroków.

Jeżeli zastanowimy się nad tym, w jaki sposób postać się porusza, będziemy w stanie rozłożyć czynności potrzebne do spacerowania na poszczególne elementy składowe. Wybranie czterech wariantów ruchu duszka Avery pozwala na zaprogramowanie prostej animacji spaceru:

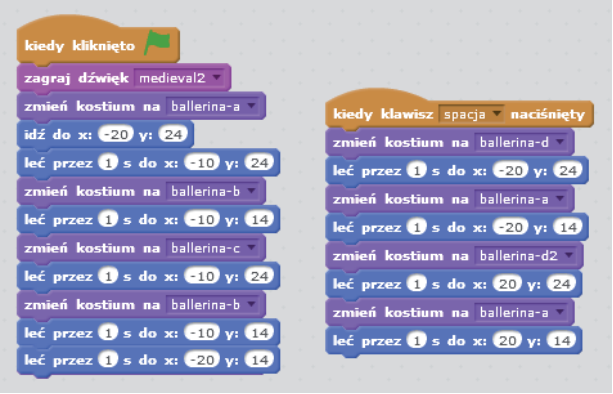


Tym razem korzystamy z granatowego bloczka „leć" i osi współrzędnych odpowiadającej za orientację na scenie. Ciemnofioletowe tło pozwala nam zaprogramować kostiumy, czyli kolejne fazy ruchu Avery. Aby spacer duszka był nieco dłuższy, używamy bloczków „sklonuj siebie" i „kiedy zaczynam jako klon" dostępnych w żółtym polu „Kontroli". Ukrywając pierwszą postać Avery w miejscu oznaczonym współrzędnymi „- 120, 0", sprawiamy, że duszek rozpoczyna dalszą część spaceru od kolejnego oznaczonego przez nas miejsca w sposób

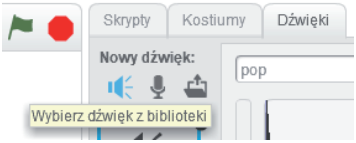
niedostrzegalny dla osoby obserwującej animację. Proponujemy przećwiczyć inne warianty ruchu, korzystając z dostępnych w bibliotece duszków.

Ćwiczenie 3: Taniec.

Aby program był ciekawszy, możemy dodać do niego dźwięk i zaprosić na scenę cztery kostiumy duszka – baletnicy:

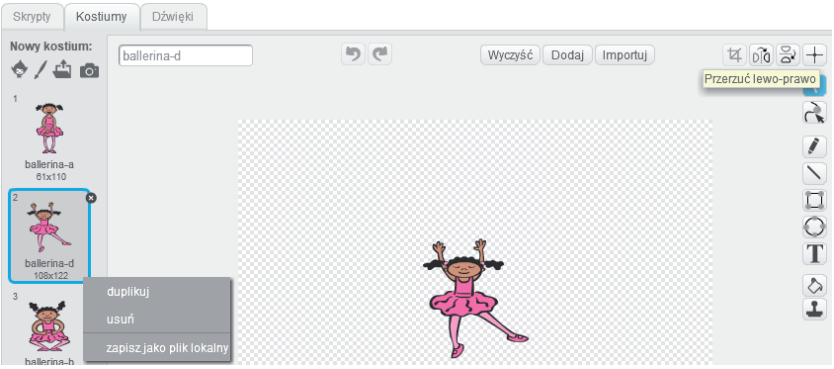


Zakładka „Dźwięki" umożliwia nam wybranie dźwięku z pliku, nagranie własnego dźwięku lub skorzystanie z dźwięków dostępnych w bibliotece:



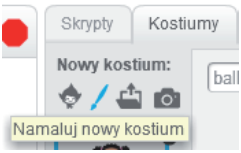
Możemy manipulować dostępnymi dźwiękami, korzystając z bloczków na jasnofioletowym tle.

Ten program jest nieco bardziej interaktywny, ponieważ użytkownik może sam ustawić wykonanie skryptu odpowiadającego za podskoki, naciskając klawisz spacji. Aby podskok mógł być wykonany równocześnie w lewo i w prawo, stworzyliśmy kostium „ballerina-d2" dostępny w zakładce „Kostiumy". Należy wyszukać kostium „ballerina-d", PPM rozwinąć menu umożliwiające duplikowanie duszka, a następnie skorzystać z opcji „przerzuć lewo-prawo".



Proponujemy poćwiczyć wykorzystanie innych dostępnych dźwięków przy użyciu jasnofioletowych bloczków „Dźwięki".

To również dobry moment, by poćwiczyć rysowanie własnych duszków:



4.3. Rysowanie figur geometrycznych – konstruowanie prostych skryptów

Ćwiczenie 4: Rysowanie kwadratu.

Bloczki na ciemnozielonym tle („Pisak") odpowiadają za rysowanie.

Wstawiając do skryptu bloczek „przyłóż pisak", a następnie przesuwając duszka, uzyskujemy efekt rysowania linii. Na początku skryptu wstawiliśmy bloczki „wyczyść" oraz „idź do", porządkujące wygląd sceny w pierwszej fazie działania programu. Następnie, po raz pierwszy sięgamy po pętlę „powtórz … razy", dostępną na żółtym tle „Kontroli", której użycie znacząco skraca skrypt. W tym przypadku 8 linii, „przesuń" i „obróć", zastępujemy dwoma liniami koniecznymi do narysowania kwadratu.

Proponujemy przećwiczenie rysowania innych figur i zmodyfikowanie skryptu w celu uzyskania programu rysującego inne figury geometryczne, takie jak: prostokąt, trójkąt, trapez, romb i koło.

Mogą Państwo również eksperymentować z użyciem następujących bloczków: „ustaw kolor pisaka", „zmień kolor pisaka", „zmień odcień pisaka" oraz „zmień rozmiar pisaka".

4.4. Doskonalenie umiejętności pisania, czytania, dodawania, odejmowania i mnożenia

Aplikacja Scratch, oprócz poruszania duszkami, daje również możliwość projektowania interakcji człowiek - komputer.

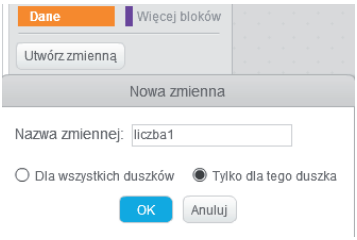
Ćwiczenie 5: Rozmowa.

Bloczki na ciemnofioletowym tle pozwalają duszkowi „mówić" i „myśleć" podobnie, jak robią to bohaterowie komiksów.

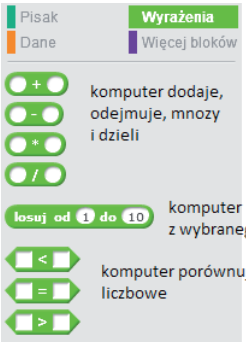
Jasnoniebieskie „Czujniki" umożliwiają komunikację z duszkiem poprzez klawiaturę.

Ćwiczenie 6: Dodawanie.

Pomarańczowe bloczki („Dane") pozwalają nam przechowywać informacje o liczbach w zmiennych.

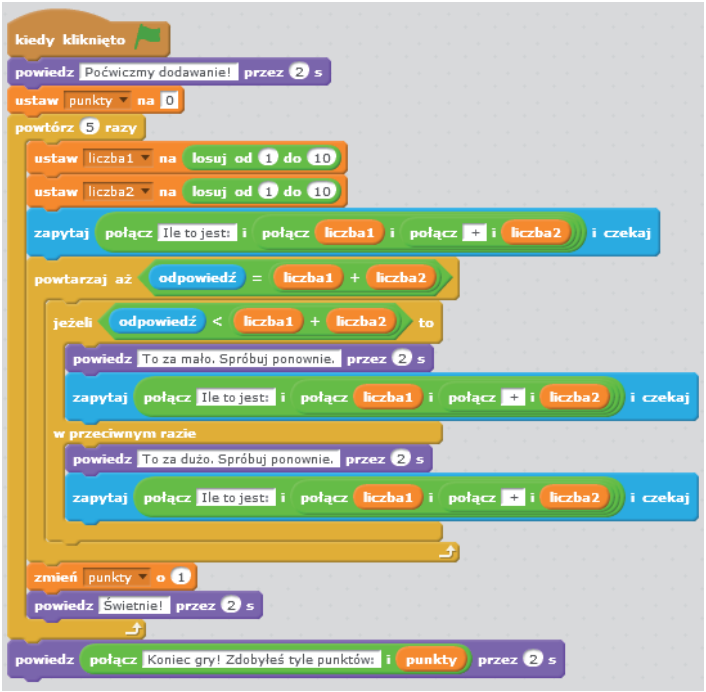


Do tej pory korzystaliśmy z jasnozielonego tła „Wyrażenia", aby połączyć słowa zaprogramowane w skrypcie ze słowami wpisanymi za pomocą klawiatury („połącz… i …"). Pozostałe bloczki „Wyrażeń" to raj dla matematyków.



Korzystając z dotychczasowej wiedzy oraz bloczków wyrażeń można stworzyć grę, w której uczniowie będą ćwiczyć dodawanie. Poeksperymentujmy w niej z nowymi rodzajami pętli: „jeżeli… to", „jeżeli/ w przeciwnym razie" oraz „powtarzaj aż". Możliwe jest tworzenie różnych wariantów skryptu, co dowodzi, że programowanie pomaga w rozwijaniu dywergencyjnego i twórczego myślenia, jak również kreatywności i wyobraźni.

Przykład rozbudowanej wersji gry przedstawiono poniżej.



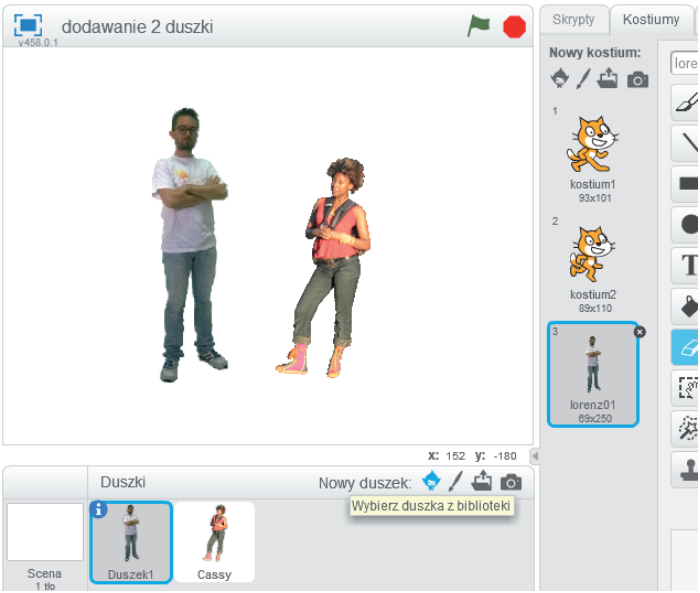
Proponujemy wykonanie kilku ćwiczeń z użyciem pętli, wstawiając je w różne miejsca skryptu w odmiennych konfiguracjach.

Można też zmodyfikować skrypt tak, aby zamiast dodawania umożliwił ćwiczenie tabliczki mnożenia.

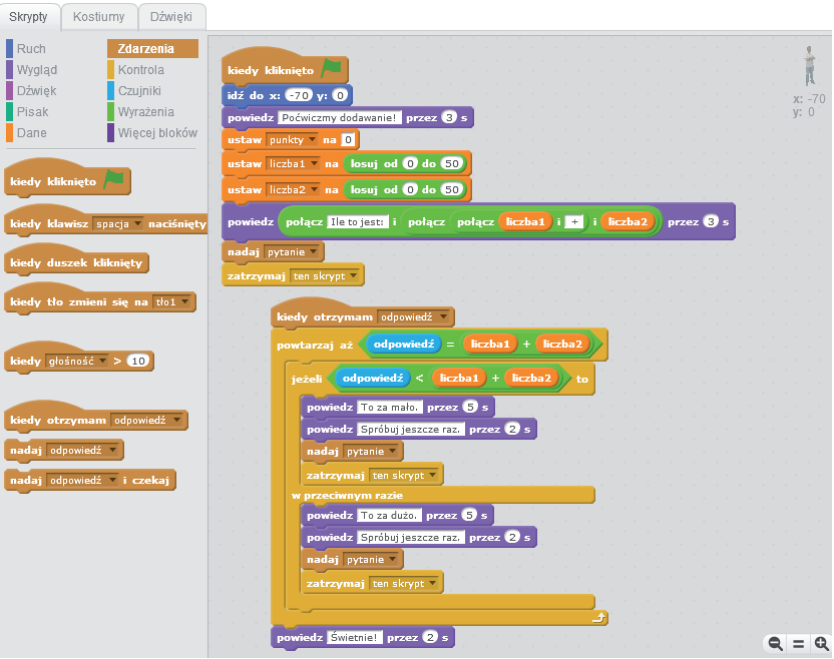
Ćwiczenie 7: Rozmowa dwóch duszków.

Dodawania może być jeszcze ciekawsze, gdy na ekranie będziemy śledzić interakcję dwóch duszków, jednocześnie wcielając się w jednego z nich.

Tworzymy program, wybierając dwa duszki.

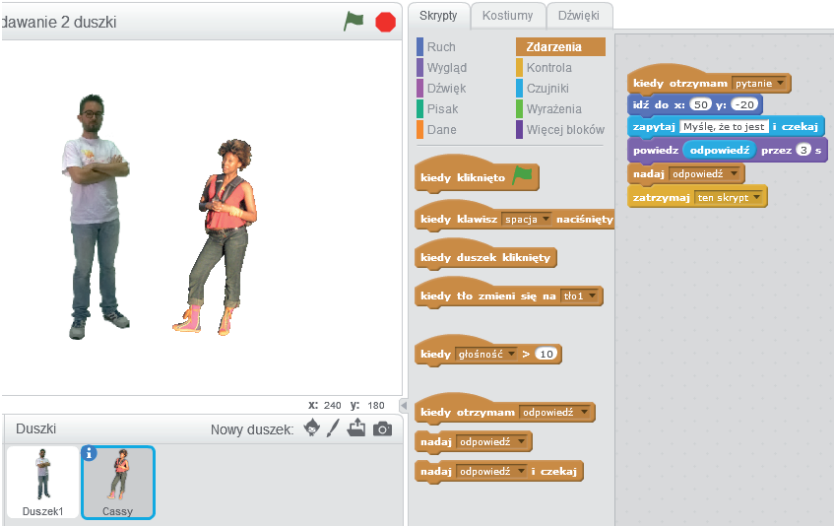


Następnie piszemy skrypty, osobno dla każdego duszka. Skrypty 1 duszka (Lorenzo):

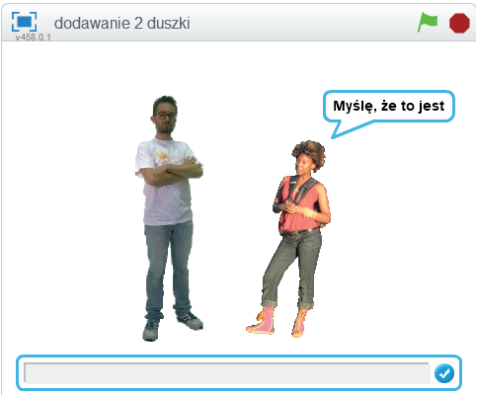


Program uruchamia się po kliknięciu na zieloną flagę. Pierwszy duszek zadaje pytanie i przechodzi w stan oczekiwania („zatrzymaj ten skrypt"). Uaktywnia się ponownie, gdy otrzyma odpowiedź od drugiego duszka („kiedy otrzymam odpowiedź"). Sprawdza, czy odpowiedź jest poprawna i chwali za jej udzielenie. Jeżeli odpowiedź jest niepoprawna, uaktywnia się pętla w pętli - kiedy odpowiedź jest niższa niż suma wylosowanych liczb, pierwszy duszek podpowiada, że to za mało, w przeciwnym razie podaje komunikat „To za dużo". Zarówno w przypadku zbyt małej, jak i zbyt dużej liczby, duszek 1 przechodzi w stan oczekiwania na kolejną odpowiedź.

Wybór komunikatu „pytanie” sprawia, iż rozpoczyna działanie drugi duszek (Cassey):



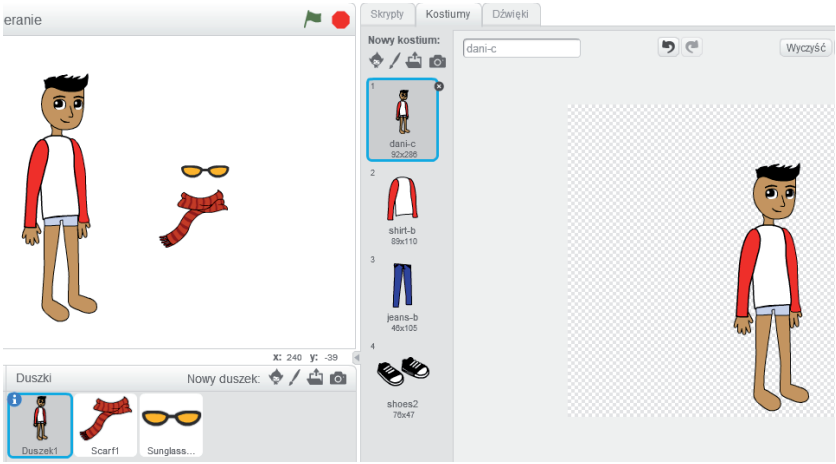
Użytkownik wciela się w rolę drugiego duszka, wpisując na klawiaturze odpowiednią odpowiedź. Duszek 2 wyświetla tę odpowiedź i nadaje ją jako komunikat przywołujący do działania duszka 1. W tym czasie duszek 2 przechodzi w stan oczekiwania. Jeżeli jego odpowiedź jest poprawna, nie musi już nic więcej robić. Jednakże w przypadku błędnej odpowiedzi (liczba jest zbyt mała lub zbyt duża), użytkownik zostanie ponownie poproszony o wpisanie poprawnej liczby.



Odpowiedź wpisujemy za pomocą klawiatury i zatwierdzamy klawiszem „enter” lub niebieskim znaczkiem na ekranie, znajdującym się z prawej strony wpisanej przez nas liczby.

Ćwiczenie 8: Pogoda.

W tym ćwiczeniu duszek Dani-c zastanawia się, co na siebie włożyć. Wybieramy z biblioteki odpowiednie ubrania: shirt-b, jeans-b, shoes2 oraz dwa kolejne duszki: sunglasses2 i scarf1. Przeciągamy koszulę, dżinsy i buty na postać duszka.



Skrypt pierwszego duszka wygląda następująco:

Skrypt duszka „okulary" zaczyna działać, gdy za pomocą klawiatury wpiszemy odpowiedź „tak", oznaczającą, że jest ciepło.

Jeżeli wpisaliśmy inną odpowiedź, uruchamia się skrypt duszka „szalik"

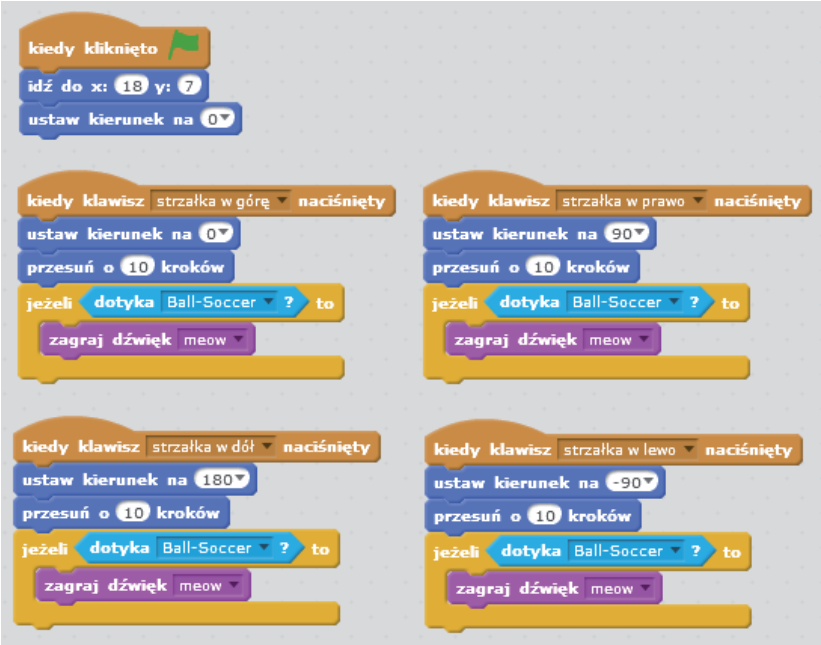
Ćwiczenie 9: Gra.

Na zakończenie rozdziału poświęconego pracy w środowisku Scratch proponujemy prostą grę, w której kotek goni piłkę.

Potrzebujemy dwóch graczy i dwa duszki - kota i piłkę.

Jeden gracz steruje piłką za pomocą myszki. Skrypt duszka – piłka jest bardzo krótki i działa, dopóki nie zakończymy programu czerwonym przyciskiem (stop), znajdującym się obok zielonej flagi u góry sceny.

Drugi gracz steruje ruchami kota, używając w tym celu strzałek. Skrypty duszka – kota wyglądają następująco:



Życzymy owocnych eksperymentów i dobrej zabawy z programowaniem wizualnym!

5. Robotyka

5.1. Roboty Dash & Dot

5.1.1. Wprowadzenie w świat robotyki i programowania

Programowanie to między innymi rozwijanie umiejętności analitycznego myślenia oraz rozwiązywania problemów. Opracowanie nawet najprostszego programu wymaga od „konstruktora" rozłożenia zadania na czynniki pierwsze, zrozumienia, z jakich kroków się składa i w jakiej kolejności należy je wykonać. Osiągnięcie zamierzonego celu odbywa się niejednokrotnie dopiero w kolejnej próbie. Za każdym razem należy sprawdzić, czy wszystko działa poprawnie i skorygować ewentualne błędy. Taką procedurę powinniśmy powtarzać aż do skutku, ucząc się wytrwałości w dążeniu do celu.

W programowaniu bardzo ważna jest kreatywność, bo każdy problem ma wiele rozwiązań, które prowadzą do zamierzonego celu. Nawet jeśli w danej chwili zaproponowane rozwiązanie jest właściwe, już po chwili może się okazać, że istnieją jeszcze inne, doskonalsze sposoby.

Planowanie i logiczne myślenie to kolejne umiejętności, które rozwijane są w czasie nauki programowania. Napisanie programu wymaga od programisty zaplanowania kilku kroków do przodu. Refleksja nad krokami, które zostały wcześniej zaplanowane, pozwala na osiągnięcie wyznaczonego celu.

Programowanie daje również możliwość świadomego korzystania z nowych technologii. Poznajemy, w jaki sposób coś działa, mamy więc szansę nauczyć się to naprawiać i udoskonalać.

Jednakże głównym celem nauki programowania jest możliwość tworzenia programów na ekranie komputera czy tabletu i przekładanie tego na rzeczywistość. Napisany program służy bowiem do tego, żeby coś się zdarzyło w świecie realnym i może mieć wpływ na działalność człowieka.

Bardzo dobrym narzędziem do nauki programowania poprzez zabawę i pokazania, że stworzenie programu nie kończy się na ekranie komputera, są Dash i Dot. Roboty wzbudzają bardzo pozytywne emocje. Wykonują one „na żywo" polecenia zaprogramowane na ekranie komputera.

Roboty świetnie nadają się do rozwijania umiejętności współpracy i komunikacji podczas pracy w małych grupach. A właśnie te umiejętności to kluczowe kompetencje XXI wieku, przydatne nie tylko przyszłym informatykom czy programistom, ale również w każdym innym zawodzie oraz w codziennym życiu.

5.1.2. Budowa robotów, montowanie akcesoriów.

Roboty Dash i Dot wykorzystywane są przede wszystkim jako pomoce dydaktyczne. Sprawdzają się głównie na początkowym etapie nauczania programowania, w przedszkolach i szkołach podstawowych. Z robotów można korzystać nie tylko na zajęciach komputerowych, ale również w trakcie innych zajęć lekcyjnych do wizualizacji opracowanych zagadnień nauczania.

Zestaw podstawowy składa się z dwóch robotów Dash i Dot.

Dash to robot „dynamiczny", który może poruszać się z dowolną prędkością w różnych kierunkach dzięki dwóm kółkom umieszczonym pod spodem. Porusza on również głową i wydaje dźwięki, łącznie z tymi, które nagra użytkownik. Dzięki wbudowanym mikrofonom robot usłyszy klaśnięcia lub zareaguje na kierunek, z którego do niego mówimy. Zainstalowane światła dają możliwość świecenia, a czujniki pozwalają na wykrywanie przeszkód znajdujących się z przodu lub z tyłu robota.

Dot jest nieruchomym elementem zestawu, ale dzięki akcelerometrowi rozpoznaje, czy jest potrząsany, przechylony lub czy został porzucony. Ma światła i mikrofon, może też wydawać dźwięki tak jak Dash. Po odłączeniu od podstawy, staje się kulą, którą można toczyć.

DODATKI I AKCESORIA

1. Łączniki do klocków LEGO - mogą być przytwierdzone do dowolnego z 6 punktów łączenia Dasha, znajdujących się na głowie i ciele robota. W tym celu należy przyłożyć łącznik okrągłym otworem (strona wypukła) do punktu łączenia na robocie. Można je zamontować pod kątem 90 lub 45 stopni (tj. pionowo w górę lub w dół, do przodu, do tyłu czy ukośnie).

2. Spychacz montowany jest do punktów łączenia, znajdujących się na dwóch przednich kołach Dasha.

3. Uszy Króliczka można przyczepić zarówno do głowy, jak i ciała robota. W celu zamontowania, należy je przyłożyć okrągłym otworem (strona wypukła) do punktów łączenia na Dashu.

4. Wyrzutnia piłeczek.

5. Uchwyt na smartfona pozwala na robienie zdjęć i kręcenie filmów z perspektywy robota.

6. Uchwyt do holowania to przydatne narzędzie, które daje robotom wiele nowych możliwości przemieszczania się.

7. Cymbałki.

5.1.3. Instalacja aplikacji na tablecie. Konfiguracja i ładowanie robota.

Roboty Dash i Dot współpracują z tabletem lub smartfonem wyposażonym zarówno w system Android 4.4.4 lub nowszy, jak i iOS 8.0 lub nowszy czy z zainstalowanymi bezpłatnymi aplikacjami Wonder Workshop. Aplikacje te można pobrać z App Store lub Google Play. Całość oprogramowania składa się z pięciu aplikacji dostosowanych dla każdego, niezależnie od wieku czy poziomu wiedzy. W skład zestawu aplikacji wchodzą: Go, Path, Xylo, Wonder, Blockly. Instalacja aplikacji odbywa się standardowo. Po pobraniu, należy uruchomić (jeśli nie nastąpi to automatycznie) kreatora instalacji. W trakcie pracy z aplikacjami nie jest wymagany dostęp do sieci Internet, jednakże w czasie pierwszego uruchamiania aplikacje sprawdzają, czy są dostępne aktualizacje. Od czasu do czasu będzie się tak działo przy kolejnym uruchamianiu. Aktualizacja oprogramowania może trwać od 3 do 20 minut i jest niezbędna do poprawnego działanie robotów i wprowadzania nowych funkcji.

Uruchomienie robotów jest bardzo proste. Wystarczy wcisnąć przycisk Power. Fabrycznie urządzenia mają wgrane oprogramowanie, które pozwala im poruszać się, świecić i wydawać dźwięki. Należy pamiętać o doładowaniu wbudowanych akumulatorów. W tym celu podłączamy robota za pomocą kabla (wchodzącego w skład zestawu) do komputera lub wykorzystujemy standardową ładowarkę do smartfona lub tabletu. Pełen cykl ładowania trwa ok. 60-90 minut.

Połączenie robotów z urządzeniem mobilnym odbywa się za pomocą Bluetooth 4.0. Możemy nawiązać kontakt z robotem w wybranej aplikacji poprzez naciśnięcie plusa w górnym rogu. Możliwa jest też komunikacja z kilkoma robotami przy użyciu urządzenia mobilnego. Natomiast jeden robot może być sterowany z jednego smartfona lub tabletu.

W trakcie uruchamiania robotów istnieje możliwość ich personalizacji, poprzez nadanie im indywidualnych imion, kolorystyki czy sposobu świecenia diod.

5.1.4. Charakterystyka dodatkowych aplikacji Go, Path, Wonder, Blockly, Xylo.

Aplikacje, które można wykorzystać do nauki programowania, dostępne są do bezpłatnego pobrania w App Store, Google Play lub na stronie internetowej makewonder.pl. Intuicyjny interfejs sprawia, że aplikacje te są bardzo przyjazne dla młodych programistów. Istnieją też aplikacje innych firm dla robotów Dash i Dot, np. Swift Playgrounds lub Blocklify, dostępne w polskiej wersji językowej oraz SWIFT Playgrounds na iPady.

W skład pełnego zestawu aplikacji Wonder Workshop dla robotów Dash i Dot wchodzą: Go, Path, Wonder, Blockly i Xylo. Według producenta, robotami mogą bawić się dzieci od 5 roku życia. Należy jednak liczyć się z tym, że 5-6 latki będą korzystać z aplikacji Go, Path lub Xylo. Natomiast Blockl i Wonder powinny być proponowane starszym uczniom, którzy posiadają już podstawowe umiejętności związane z programowaniem robotów Dash i Dot.

Go - podstawowa aplikacja, dzięki której programista zapoznaje się z funkcjami robotów. Jest to panel sterowania, umożliwiający zdalne sterowanie robotami, zmianę kolorów światełek obu robotów, odgrywanie dźwięków oraz nagranie własnego głosu. Nie występują tu jeszcze elementy programowania.

Path - należy do mniej skomplikowanych aplikacji. Dzięki interfejsowi i scenariuszowi przypominającemu grę, może być traktowana jako wstęp do elementów programowania dla najmłodszych uczniów. Po prawidłowym narysowaniu trasy (sekwencja algorytmu), robot wykona określone polecenia. Zadaniem ucznia jest wyznaczenie miejsca, w którym ma się znaleźć Dash, a następnie narysowanie odpowiedniej trasy poruszania się robota oraz przeciągnięcie na trasę wybranych komend dostępnych w programie lub nagranych przez użytkownika. Zaleca się wykonanie zadań zawartych w samouczku, co ułatwi prawidłowe posługiwanie się komendami – poleceniami przeznaczonymi dla Dasha.

Blockly - aplikacja zalecana do nauczania programowania w przypadku dzieci, które posiadają podstawową znajomość języka angielskiego. Aplikacja bardzo przypomina Scratcha, gdyż programowanie robota polega na łączeniu klocków z komendami w schematy blokowe. Układanie klocków to nic innego jak tworzenie komend, tj. poleceń dla robota, które ma wykonać krok po kroku. Jest to najłatwiejszy, a co za tym idzie bardzo przyjazny, sposób tworzenia algorytmów i wykonania poleceń przez robota. Dodatkowo można wykorzystać klocki aktywatory, czyli czynniki wywołujące dane polecenia, np. klaśnięcie w dłonie będzie powodowało ruch robota wzdłuż linii prostej na dystansie 30 cm. Istnieje też możliwość tworzenia pętli, warunków czy bloków zmiennych.

Xylo - specjalna aplikacja dla Dasha i osobnego akcesorium – cymbałek. Za pomocą Xylo programista komponuje proste melodie, które zagra Dash. Kolory w aplikacji są odpowiednikiem klawiatury dzwonków (gama C-dur), dzięki czemu mogą być zapisane w postaci nut. Przeznaczona jest dla dzieci od 8 roku życia, ale mogą z niej korzystać młodsi.

Wonder - najbardziej rozbudowana aplikacja obrazkowa do nauki programowania. Program tworzy się z dostępnych ikon, łącząc je za pomocą linii. Można też dodawać zdarzenia. Aplikacja dostępna jest w języku angielskim, ale nie powinno to stanowić problemu, gdyż jest bardzo intuicyjna i obsługiwana wizualnie. Programując roboty, wymuszamy na nich odpowiednie zachowania, jak również wykorzystujemy dźwięk, ruch i światło.

Wonder składa się z trzech programów:

1. Scroll Quest – zawiera kilkadziesiąt zadań do rozwiązania dla Dash i Dota.
2. Free Play – plansza, na której można tworzyć własne programy.
3. Controller – panel sterowania robotami (podobny jak w aplikacji Go), wzbogacony o instrukcję obsługi wyrzutni piłeczek.

5.1.5. Zapoznanie z możliwościami robotów Dash i Dot

Roboty Dash i Dot są przykładem maszyn zbierających informacje na temat środowiska, w którym mają wykonać „zlecone" przez programistę polecenia. Reagują dzięki czujnikom, a otrzymane informacje przetwarzają w programie i postępują zgodnie z poleceniami, wykonując określone działania. Dash dla przykładu wyczuwa obiekty w otoczeniu, reaguje na naciśnięcie przycisków znajdujących się na głowie robota oraz na usłyszane dźwięki. Dot również reaguje na naciśnięcie przycisków umieszczonych na głowie czy na dźwięki, ale dodatkowo również na wstrząsy, zmianę wysokości lub toczenie nim. W przypadku robota Dash mamy też możliwość zapisywania informacji i planowanie dalszych działań.

Jednakże należy pamiętać, iż oba roboty wykonują tylko te działania, czynności lub reakcje na określone sytuacje, do których podamy wyraźną instrukcję zapisaną w formie programu. Program jest więc instrukcją krok po kroku mówiącą komputerowi (w naszym przypadku robotowi), co robić i jak reagować na określone bodźce.

4.1.6. Posługiwanie się poleceniami typu: press and hold, personalize, greeting, forward, light, sound.

Press and hold, personalize, greeting, forward, light, sound to podstawowe komendy, dzięki którym roboty Dash i Dot będą wykonywały wszelkie czynności zaprogramowane przez użytkownika. Komendy te, będące słownictwem bazowym, znajdziemy we wszystkich aplikacjach obsługujących roboty.

- **Press and hold** – naciśnij i przytrzymaj – to polecenie dla programisty, który łączy robota ze swoim robotem po raz pierwszy. Komenda służy do nawiązania łączności pomiędzy panelem roboczym (aplikacją) a robotem.
- **Personalize** – personalizacja robota, polegająca na dostosowaniu wyglądu robota, sposobu świecenia światełek, wydawania dźwięku czy nawet nadaniu indywidualnego imienia robotowi.
- **Greeting** – powitanie – polecenie, dzięki któremu robot będzie się witał z użytkownikiem podczas włączania oraz podczas nawiązywania łączności z programistą.
- **Forward** – naprzód – komenda dla robota, aby przemieszczał się do przodu, na określoną przez programistę odległość, z zaprogramowaną szybkością (bloczki ruchu).
- **Light** – zapalać – polecenia, które znajdują się w bloczkach programujących świecenie diod robotów. Bloczki te nadają atrybuty sposobu świecenia i koloru.
- **Sound** – dźwięk – atrybuty dźwiękowe, z których możemy wybierać własne dźwięki (nagrane), powitania, odgłosy zwierząt lub dźwięki pojazdów.

5.1.7. Ćwiczenia umożliwiające ruch robotów, omijanie przeszkód, tworzenie toru jazdy przypominającego figury geometryczne, doskonalenie umiejętności dodawania, odejmowania, mnożenia oraz pomiaru odległości.

Nauka z Dashem – poruszanie się robota[5]

Pomoce:

- plansze do algorytmiki;
- tablety;
- zestaw Dash i Dot;
- plansze ze znakami drogowymi STOP, ustąp pierwszeństwa, jesteś na drodze z pierwszeństwem, kolorowe kartony (czerwony, żółty, zielony), przejście dla pieszych, ścieżka rowerowa, ścieżka dla pieszych, zakaz wjazdu, nakaz jazdy w prawo, nakaz jazdy w lewo.

Przebieg zajęć:

1. Powitanie uczniów, a następnie twórcza rozgrzewka w kręgu – przypominamy sobie, co robiliśmy na ostatnich zajęciach i czy w związku z tym mamy jakieś pytania lub ciekawe pomysły, na przykład zadania dla robotów.

[5] Autor: mgr inż. Agnieszka Krawińska - udostępnione na licencji CC BY 3.0 PL

2. Rozmowa nauczyciela z dziećmi o tym, czym się będą dzisiaj zajmować – nauczymy Dasha bezpiecznie jeździć po placu rowerowym. Pytamy, jakie znaki drogowe dzieci znają, a następnie czy wiedzą, co oznaczają znaki, które przygotowaliśmy. Dzieci odpowiadają, nauczyciel wyjaśnia konieczność przestrzegania przepisów ruchu drogowego.
3. Dzieci wraz z nauczycielem układają plac rowerowy z akcesoriów znajdujących się na sali – mogą to być książki, ołówki czy pudełka. Dzieci wspólnie planują plac w formie labiryntu oraz przecinających się ścieżek i umieszczają znaki drogowe w odpowiednich miejscach.
4. Plac powinien posiadać 3 wjazdy/wyjazdy. Zadanie Dasha polega na wjechaniu na plac jednym wjazdem i wyjechaniu innym, jak również na przestrzeganiu wszystkich napotkanych przepisów drogowych.
5. Nauczyciel jeszcze raz objaśnia zasady poruszania się w labiryncie:
    a. jeżeli Dash zobaczy znak Stop lub Ustąp pierwszeństwa, musi się zatrzymać i przepuścić robota mającego pierwszeństwo;
    b. przed pasami należy się zatrzymać i przepuścić pieszego (jeden lub dwa roboty mogą odgrywać rolę pieszego/pieszych);
    c. nie wolno wjechać w alejkę oznaczoną zakazem wjazdu ;
    d. w przypadku nakazu skrętu w prawo lub lewo, należy pojechać we wskazanym kierunku.
6. Dzieci w parach programują Dasha i sprawdzają, czy potrafi przejechać bezpiecznie przez plac.
7. Kończymy zajęcia rundą: „Dzisiaj nauczyłem/łam się, że….” „Najbardziej podobało mi się…”,„Najtrudniejsze było dla mnie…”

Wskazówki praktyczne:

Jeżeli dzieci są na podobnym poziomie pod względem posiadanej wiedzy i umiejętności, możemy je podzielić na grupy w sposób losowy (losowanie karteczek z imionami dzieci, kolorowych kulek, itp.). W przypadku gdy poziom ten jest znacznie zróżnicowany, należy przydzielić uczniów do grup o podobnym poziomie – unikniemy wówczas takich sytuacji, w których dzieci bardziej zaawansowane będą się nudziły, a te z niższymi kompetencjami wyjściowymi zniechęcą się ze względu na zbyt szybkie tempo pracy. Nauczyciel powinien i obserwować grupę i zarządzać przerwy, w zależności od potrzeb uczniów.

Dash ogrodnikiem[6]

Wprowadzenie

Podczas tej lekcji z Dashem będziemy doskonalić umiejętność mierzenia boków figur geometrycznych i obliczania ich obwodów.

Cele lekcji:

- kształcenie umiejętności dokładnego i starannego kreślenia kwadratu, prostokąta i trójkąta - zgodnie z własnościami tych figur;
- zapoznanie ze sposobami obliczania obwodów figur;
- doskonalenie umiejętności efektywnej współpracy w grupie.

Przedmiot: edukacja matematyczna

Wielkość grupy: ok. 4 uczniów (dzielimy klasę na 5 grup)

Klasy: 1-3

Wymagany czas: 45 min

Co będzie potrzebne: roboty i akcesoria - Dash

Pomoce dydaktyczne:

- urządzenie mobilne z zainstalowaną aplikacją Blockly - jedno na grupę;

  miarki - po jednej na grupę;
- rzutnik i komputer;
- taśma malarska - po jednej na grupę.

Materiały do ściągnięcia:

Załącznik 1 - karta pracy

Załącznik 2 - zrzut ekranu z przykładowym kodem w aplikacji Blockly

Ekran wprowadzający i porządkujący wiadomości dotyczące obliczania obwodów figur geometrycznych na przykładzie obwodów pól uprawnych.

Przebieg zajęć:

Obwody figur geometrycznych - 10 min.

Przywitaj uczniów, przedstaw Dasha, a następnie włącz animację „Obwody figur geometrycznych." Po obejrzeniu filmu podziel uczniów na czteroosobowe grupy. Można w tym celu wykorzystać karteczki w różnych kolorach. Każda grupa siada wokół arkusza szarego papieru, który nauczyciel wcześniej przykleił taśmą malarską do podłogi.

---

[6] Autor: Monika Walkowiak; http://nauczyciele.makewonder.pl/scenariusz-dash_ogrodnik.html

Pomagamy Dashowi w ogrodzie - 35 min

Uruchamiamy Dasha, który mówi uczniom treść zagadki: "Znajdę je w zupie i w sałatce też. Co mam na myśli, na pewno już wiesz!"

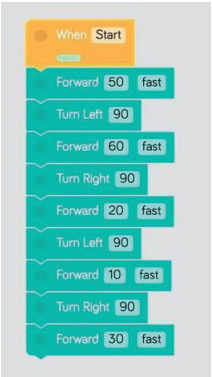
Aby to Dash powiedział wierszyk, musimy poświęć trochę czasu przed zajęciami i nagrać swój głos dla Dasha z tekstem rymowanki. W tym celu należy użyć opcji nagrywania dźwięków w aplikacji GO - nagrywamy swoją ścieżkę dźwiękową pod wybraną cyfrą, np. 1 (uwaga: mów głośno i sprawdź nagrany dźwięk, w razie potrzeby nagraj się ponownie). Możemy potem odtworzyć w klasie zapisaną ścieżkę dźwiękową (nr 1).

Dash uwielbia świeże warzywa, dlatego planuje posiać w swoim ogródku sałatę, rzodkiewkę, szczypiorek i marchewkę. Wcześniej musi jednak ogrodzić teren, aby Dot nie zniszczył jego plonów.

Rozdaj grupom karty pracy z zadaniem do wykonania. Upewnij się, że uczniowie zrozumieli treść wszystkich poleceń i poproś, aby uważnie przyjrzeli się obrazkowi na karcie. Przedstawia on pomniejszony ogródek Dasha. 2 kratki na rysunku odpowiadają 10 cm, a czerwona linia oznacza bramę wejściową do ogrodu. Zadaniem uczniów jest wyznaczenie powierzchni ogrodu za pomocą taśmy malarskiej oraz zmierzenie miarką długości poszczególnych boków.

Uczniowie pomagają Dashowi obliczyć, ile centymetrów siatki musi przygotować na ogrodzenie swojego ogródka. Przypominamy, że tam gdzie jest brama, nie będzie potrzebne ogrodzenie. Następnie uczniowie planują, w którym miejscu ogrodu Dash posieje warzywa i rysują ścieżkę dla Dasha.

Dash często upewnia się, czy jego ogród jest bezpieczny, sprawdzając stan techniczny ogrodzenia. Poproś uczniów, by w aplikacji Blockly zaprogramowali robota tak, by objechał dookoła cały ogród. Dash może wystartować z dowolnie wybranego punktu. Przykładowy program w Blockly:



Kiedy każda z grup wykona zadania, zaproponuj uczniom obejrzenie wszystkich ogrodów i wybranie najciekawszego z nich.

Rzut do celu[7]

Działania matematyczne i cyfry w aspekcie porządkowym.

Opis:

Dodawanie i odejmowanie można ćwiczyć z dziećmi na różne sposoby. Im częściej urozmaicamy zajęcia z zakresu pojęć matematycznych i prowadzimy je w formie zabawy, tym szybciej dzieci przyswoją zakres materiału, nawet o podwyższonym stopniu trudności. Na tych zajęciach, oprócz doskonalenia umiejętności dodawania, odejmowania, rozpoznawania cyfr i właściwego ich użycia w aspekcie porządkowym, dzieci trenują też rzut do celu czy dobór właściwej siły i odległości, programując robota Dash w aplikacji „Blockly".

Cele zajęć:

- udział we wspólnych zabawach;
- ćwiczenie umiejętności przymocowania w prawidłowy sposób wyrzutni piłeczek do robota oraz wkładania piłki przed rzutem;
- ćwiczenie umiejętności oszacowania odległości, dopasowania kierunku i siły rzutu;
- ćwiczenie umiejętności dodawania i odejmowania w zakresie od 1 do 9;
- ćwiczenie umiejętności właściwego używania liczb w aspekcie porządkowym;
- ćwiczenie umiejętności programowania robota w aplikacji „Blockly".

Wielkość grupy: 4 - 7 osób (dzielimy klasę na 3 zespoły)

Wiek dzieci: 5-6 lat

Wymagany czas: 1godzina

*Warto zalaminować wszystkie przygotowane kartoniki z cyframi, dzięki czemu będą one trwałe i będzie można je wykorzystać podczas innych zajęć.

**Kostkę potrzebną do zajęć można wykonać samodzielnie, wkładając dwie małe kostki do sześciennego, przezroczystego pojemniczka i naklejając na jego ściankach symbole plusów i minusów. Można również zastąpić taką kostkę dwoma zwykłymi kostkami, a rodzaj działania losować z kubeczka przy użyciu patyczków z plusami i minusami lub za pomocą karteczek z zapisanymi symbolami działań matematycznych.

Przebieg zajęć:

Wspólne przygotowanie do zajęć:

- Zgromadź w trzech miejscach sali wszystkie potrzebne materiały do zajęć.

[7] Autor: Anna Świć; http://nauczyciele.makewonder.pl/scenariusz-rzut_do_celu.html

- Rozstaw w rzędzie dziewięć pojemniczków, do których Dash będzie wrzucał piłeczką.
- Przed każdym pojemnikiem ułóż kolejno tabliczki z cyframi od 1 do 9.

– Powiedz dzieciom, że na dzisiejszych zajęciach Dash będzie próbował trafić do pojemników z zaznaczonymi cyframi, ale zanim to zrobi, potrzebuje odpowiednich akcesoriów - wyrzutni i piłeczek.

– Pokaż dzieciom, w jaki sposób prawidłowo zamontować akcesoria, jeśli będzie taka potrzeba, pomóż im przy montażu (róbcie to delikatnie, żeby nie uszkodzić żadnego z elementów).

– Połączcie roboty z aplikacją „Blockly" na urządzeniach mobilnych, a następnie omówcie, które z dostępnych komend odpowiadają za obsługę wyrzutni, Kilkukrotnie wypróbujcie, jak działa funkcja rzutu oraz w jaki sposób zmienia się natężenie i kierunek rzutu.

– Poproś dzieci, aby zgadywały, który pojemnik (licząc od lewej do prawej) wskazywany jest przez nauczyciela. Wskaż kilka pojemników.

– Wytłumacz dzieciom, że wybór pojemnika, do którego będzie celował Dash nie jest przypadkowy i będzie wynikiem działania matematycznego.

– Podziel uczniów na 3 grupy i w każdej grupie wybierz jedną osobę odpowiedzialna za przebieg pracy (staraj się zmieniać liderów grup, jednocześnie nie namawiaj do objęcia tej roli dzieci, które źle się czują w takich sytuacjach). Twoje zadanie sprowadza się teraz do dyskretnego nadzorowania i udzielania pomocy grupom.

Dodajemy, odejmujemy i rzucamy do celu.

– Jedno dziecko z każdej grupy rzuca kostką, wyrzucony plus lub minus wskazuje działanie, które należy wykonać, a dwie małe kostki decydują o wielkości składników sumy lub odjemnika i odjemnej.

– Umawiamy się z dziećmi, że jeśli wynikiem działań będzie 0, 10 lub więcej, rzut kostką jest powtarzany do momentu uzyskania wyniku mieszczącego się wzakresie od 1 do 9.

– Prawidłowo wyliczony wynik działania upoważnia dziecko do zaprogramowania rzutu do celu Dasha - do pojemnika oznaczonego taką cyfrą, która była tym wynikiem.

– Dzieci kolejno rozwiązują zadania matematyczne, a następnie programują Dasha w aplikacji „Blockly" i sprawdzają, czy uda mu się trafić piłeczką do celu.

– Każdy trafny rzut w grupie zaznaczany jest przez dzieci plusem na kartce lub odpowiednią liczbę liczmanów (guziki, patyczki, zakrętki, klocki lego itp.).

– Na zakończenie zajęć przeliczamy trafne rzuty w każdej grupie.

Programowanie robotów.

Na tych zajęciach dzieci pracują z aplikacją „Blockly" w trzech grupach. Z menu wybieramy kategorię „Accessories", w której znajdziemy komendy do Wyrzutni piłeczek (ang. Launcher). Do zaprogramowania robota w aplikacji „Blockly" potrzebne będą jedynie dwie następujące komendy:

- *Load Launcher (left/right)* – czyli Załaduj wyrzutnię piłeczką z lewej/prawej strony - ładowanie następuje poprzez skręcenia głowy Dasha w lewą lub w prawą stronę;
- *Launch with (…) % power* - czyli Wystrzel piłeczkę z mocą... % - wybieramy moc wyrzutu od 0 do 100%.

Poproś dzieci o ułożenie skryptu, a następnie sprawdź, czy nie ma w nim błędów i, w razie potrzeby, pomóż jeskorygować. Zwróć uwagę uczniów na dobór właściwej siły rzutu, pokaż im sposób zmiany natężenia siły i spróbujcie, metodą prób i błędów, jak daleko doleci piłeczka przy danym ustawieniu siły.

5.1.8. Kształtowanie umiejętności współpracy w zespole

Nowoczesne aplikacje i programy powstają dzięki współpracy interdyscyplinarnych zespołów ludzi. Właśnie umiejętność pracy w zespole jest jedną z najbardziej pożądanych kompetencji społecznych. Jednym z celów zadań realizowanych za pomocą robotów jest przygotowanie uczniów do współpracy w grupie. Dzięki wspólnej realizacji zadań uczniowie rozwijają komunikatywność i umiejętność dzielenia się wiedzą. Pokazują sobie nawzajem, jak rozwiązać dany problem, ćwiczą sztukę argumentacji, kulturę dyskusji, elastyczność, a także umiejętność radzenia sobie w trudnych sytuacjach, szczególnie w przypadku pojawienia się błędów. Bardzo ważną umiejętnością jest takie dopasowanie się do grupy, aby móc zrezygnować z własnych celów na rzecz wspólnego rozwiązania problemu. Współpracując w zespole, uczniowie powinni obiektywnie oceniać pracę innych i dawać im informacje zwrotne oraz doceniać ich starania. Poza tym dbanie o pozytywną atmosferę pracy w zespole, dzięki redukcji napięć, gaszeniu konfliktów i kompromis, aby osiągnąć cel zespołowy – to istotne elementy pracy w grupie.

5.1.9. Kształtowanie umiejętności radzenia sobie w trudnych sytuacjach

Nauka programowania od najmłodszych lat ma m.in. na celu przygotowanie ucznia do radzenia sobie w trudnych sytuacjach, kiedy np. w czasie tworzenia algorytmu pojawiają się błędy. Wykorzystywanie atrakcyjnych pomocy naukowych rozwija ciekawość poznawczą ucznia i zachęca go do skoncentrowania się na rozwijaniu umiejętności, nie pozostawiając czasu na nudę. Uczniowie w czasie zajęć kodowania dowiadują się, czym tak naprawdę jest asertywność, jak budować poczucie własnej wartości i skuteczności, w jaki sposób być w zgodzie ze sobą, jak z szacunkiem odnosić się do innych czy dbać o swoje prawa, szanując prawa innych osób. Istotnym czynnikiem kształtującym umiejętność radzenia sobie w trudnych sytuacjach jest grupowa praca uczniów. Właśnie tego typu zajęcia pozwalają uczniom identyfikować się z grupą oraz znaleźć akceptację i zrozumienie

wśród rówieśników. Wspólne rozwiązywanie problemów, a w naszym przypadku tworzenie algorytmów, powoduje zwrócenie uwagi na pozytywne czynniki, jakimi są efekty w postaci wygenerowanego kodu poleceń dla Dasha i Dota.

5.2. Roboty Lego WeDo 2.0

Pakiet szkoleniowy LEGO Education WeDo 2.0 oparty jest na standardach nauki nowej generacji, a założenia jego twórców doskonale komponują się ze zmianami w podstawie programowej. Idea LEGO® ELEGO Education, którego częścią jest WeDo2, nie polega na dostarczeniu zabawek czy pomocy dydaktycznych do szkoły. Siła LE to wykorzystanie pozytywnych skojarzeń dzieci związanych z klockami LEGO® do wspierania kreatywności, rozbudzania wyobraźni i zachęcania do uczenia się. Autorzy pragną przenieść te wartości do klasy szkolnej, aby wspierać nauczycieli w tworzeniu bardziej angażującego i sprzyjającego procesowi edukacyjnemu środowiska. Poprzez zestawy LEGO® Education możliwe jest dostarczenie dzieciom namacalnych przedmiotów obrazujących zagadnienia, o których się uczą. To zmienia sposób, w jaki przebiega proces dydaktyczny. Wprowadzanie elementów robotyki na zajęcia w szkole ma sens, jeśli traktujemy je jako całościowy proces, przemyślany od początku do końca, a zajęcia odbywają się cyklicznie. Używając nawet najbardziej zaawansowanych narzędzi programistycznych, musimy pamiętać, że nie zastąpią one nauczyciela. Sposób, w jaki korzystamy z nich wspólnie z uczniami, pełni kluczową rolę w procesie dydaktycznym, dlatego ważne jest, żeby szczególnie dokładnie opracowywać metodykę nauczania tych treści.

5.2.1. Budowa robotów, zapoznanie z elementami zestawu, pobieranie i instalowanie oprogramowania na tablet i komputer

Podstawowy zestaw Lego WeDo umożliwia budowanie robotów, w których stosowane są napędy i przekładnie, bloczek pomiaru odległości oraz bloczek położenia. Nawet dysponując tak ograniczonym zestawem czujników, nie odczuwamy ograniczeń twórczych. Przybywa bowiem rzeczywistych maszyn na bazie wybranych czujników, co stanowi zachętę do odnajdywania nowych zastosowań. Na każdym etapie istnieje możliwość rozszerzania zestawu o elementy bierne (klocki) i czynne, czyli kolejne jednostki napędowe lub czujniki. W tym przypadku ograniczeniem jest tylko wyobraźnia. „Mózgiem" w profesjonalnych rozwiązaniach jest komputer(y) programowalny(e) WeDo Smart Hub, przekazujący polecenia do elementów wykonawczych.

W przypadku Lego WeDo, programowanie dokonywane jest przy użyciu aplikacji APP lub instalacyjnego pakietu pobieranego z firmowej strony internetowej, w zależności od ver. Windows, czy OS, za pomocą bezprzewodowego łącza Bluetooth. Komputer Lego WeDo2 nie posiada funkcji odtworzenia ostatniego programu z wewnętrznej pamięć i po wyłączeniu zasilania, program należy ponownie uruchomić. Do zasilania WeDo 2.0 wymagane są dwie baterie lub akumulatory AA lub dedykowane akumulatory 45302.

Kompatybilność oprogramowania

Systemy operacyjne, które mogą być użyte:

iOS (od 8.1) – minimalne wymagania to iPad III generacji lub iPad mini 2 – oprogramowanie do pobrania z iTunes.

Android (od wersji 4.4.2) – urządzenie musi posiadać przynajmniej BT 4.0 LE oraz ekran o przekątnej 8" lub większy. Lista urządzeń przetestowanych przez LEGO Education znajduje się na oficjalnej stronie z wymaganiami sprzętowymi. Oprogramowanie do pobrania w sklepie Google Play.

Windows 7 – pod warunkiem, że sprzęt posiada sterowniki pozwalające na współpracę z urządzeniami BT 4.0 LE – obecnie działa w 100% jedynie radio BlueGiga BLED 112 – do pobrania z LERO (po zalogowaniu na LEGOid). Uwaga - z powodów technicznych Windows 7 obsługuje połączenie tylko z jednym hubem WeDo 2.0 w danym momencie.

Windows 8.1, **10**, **10 mobile** – do pobrania ze sklepu Windows. Uwaga – Windows 10 w wersji min. 10.0.10586.420.

Mac OS (od 10.10) – do pobrania z LERO (po zalogowaniu na LEGOid). Chromebook (min 4 GB RAM, BT 4.0, 2GB dysku, procesor Intel® Celeron® 2955U lub odpowiednik, system ChromeOS w wersji 5.0 lub nowszej, połączenie z Internetem). Aplikacja do pobrania ze sklepu Chrome Web. Do działania potrzebny jest Bluetooth 4.0 (low-energy) lub nowszy.

Zestaw Lego WeDo dostarczany jest w bardzo praktycznym opakowaniu z tworzyw sztucznych.

Zespół lub uczeń, który rozpakowuje nowy zestaw, musi pamiętać o umieszczeniu spisu części w danej przegródce i precyzyjnym rozłożeniu wypakowanych z woreczków klocków. Elementy napędu i czujniki warto przechowywać w odzyskanych woreczkach PCV i trzymać „pod ręką", ponieważ są one bardzo często używane. W przypadku częstego korzystania z zestawu, uczniowie nabiorą wprawy w szybkim wyborze i dopasowywaniu elementów do projektowanej konstrukcji.

Liczne materiały poglądowe, w wersji książkowej i elektronicznej, pozwalają na zapoznanie się z Zestawem Lego WeDo. Niezwykle ważne jest, aby odpowiednio dopasować oprogramowania do posiadanego sprzętu. Zestawy Projektu IRC zostały dobrane poprawnie (Tablety, Android ver. I, Bluetooth >4.0) i nie przysparzają problemów z kompatybilnością. Informacje na temat innych rozwiązań znajdują się na stronie producenta https://education.lego.com. Odpowiednie oprogramowanie instalujemy po pobraniu ze strony https://education.lego.com/en-us/downloads/wedo-2/software.

Oprogramowanie zapewnia łatwe w użyciu środowisko programistyczne, które sprawia, że modele „ożywają", zgodnie z zaplanowanymi przez uczniów czynnościami. Oprogramowanie Lego WeDo 2 wspiera ucznia w jego działaniach, dostarczając bardzo szczegółowych instrukcji (3D), a w raz z nabywanym doświadczeniem uczeń jest w stanie wykonywać pracę samodzielnie i twórczo. Ważnym elementem oprogramowania jest zintegrowane narzędzie do tworzenia dokumentacji, za pomocą którego uczniowie mogą:

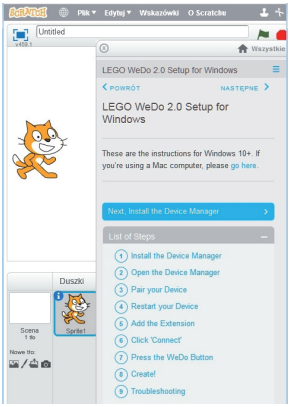
– robić zdjęcia ważnych etapów tworzenia prototypów lub ukończonych modeli;

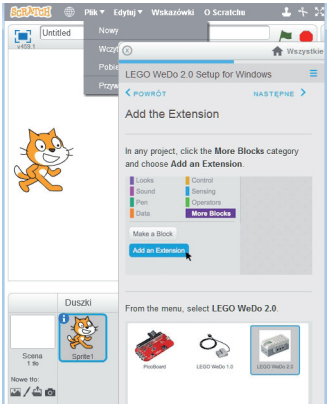
– robić zdjęcia i filmy dokumentujące pracę zespołu nad rozwiązywaniem problemów i dochodzeniem do optymalnych rozwiązań;

– opisywać w formie tekstu zjawiska, których nie da się przedstawić obrazem.

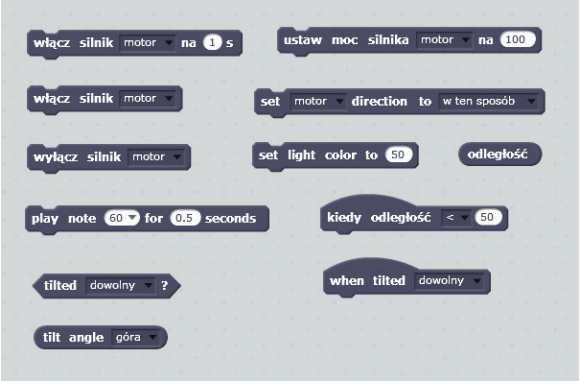
5.2.2. Dodawanie rozszerzenia w programie Scratch do obsługi Lego WeDo2

Interesującą właściwością Lego WeDo2 jest możliwość sterowania zbudowanych robotów przy użyciu aplikacji Scratch oraz tworzenia animowanych scen. Wymaga to zainstalowania „Managera Urządzeń" ze strony https://scratch.mit.edu/wedo/, który pozwala na podłączenie WeDo 2.0 do Scratcha za pomocą Bluetootha. *Uwaga! Aktualnie tylko z systemem operacyjnym Mac OSX i Windows 10+.*



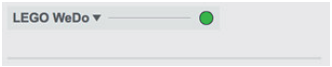


W obszarze skryptów „Scratcha" wybieramy „Więcej BLOKÓW" i dodajemy rozszerzenie WeDo2 Powinniśmy uzyskać bloczki dedykowane LEGO, jak na rysunku poniżej.

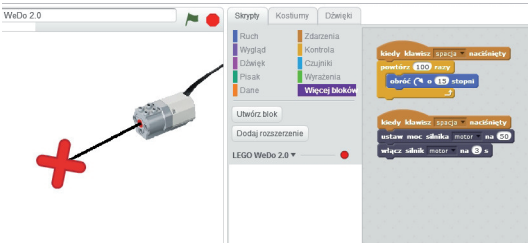


5.2.3. Zapoznanie z dostępnymi blokami w oprogramowaniu (sterowanie programem, dane z czujników, wyświetlanie czujników, dane wejściowe)

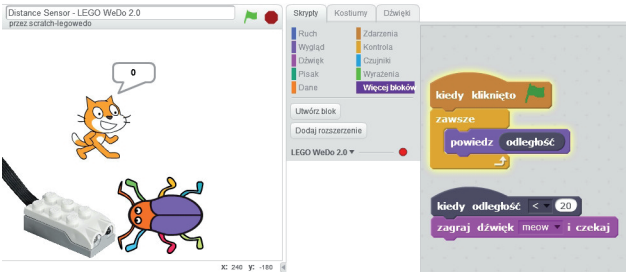
Prawidłowe podłączenie Lego WeDo2 sygnalizowane jest barwnym punktem w obszarze skryptów .



Kolor czerwony wskazuje na brak powiązanych urządzeń, kolor zielony natomiast sygnalizuje zestawienie urządzeń przez bluetooth, czyli poprawnie. Należy również pamiętać, że użycie bloczków wspierających Lego WeDo2, nie spowoduje widocznego działania na ekranie komputera. Aby je uzyskać, należy wykonać (ułożyć) zestaw bloczków dedykowanych dla Scratch. STEROWANIE:



DANE Z CZUJNIKÓW:



WYŚWIETLANIE CZUJNIKÓW I DANYCH WEJŚCIOWYCH:

5.2.4. Korzystanie z zasobów Internetowych wspierających prowadzenie zajęć

Popularność Lego WeDo2 systematycznie rośnie, a wraz z nią chęć dzielenia się wiedzą i doświadczeniem na temat prowadzonych zajęć. Warto skorzystać z podpowiedzi innych użytkowników, zanim sami staniemy się autorami metodycznych opracowań. Na początek dobrze jest obejrzeć filmy instruktarzowe na Yutube, wpisując „Yutube Lego WeDO2" - przykładowy adres https://youtu.be/vfORfOuovy4.

Jedną z bardziej przydatnych stron internetowych jest strona oficjalnego dystrybutora rozwiązań LEGO® Education w Polsce, dostępna pod adresem: https://www.akcesedukacja.pl/kontakt/. Znajdziemy tam liczne przykłady i wskazówki, możemy też dołączyć do grup dyskusyjnych.

https://www.facebook.com/groups/MistrzowieLEGO/

https://education.lego.com/en-us/support/wedo-2/building-instructions

Pamiętajmy też o otwartych zasobach edukacyjnych:

http://otwartezasoby.pl/scholaris-portal-wiedzy-dla-nauczycieli/;

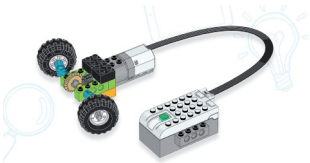
https://downloads.scratch.mit.edu/devicemanager/ScratchDeviceManager-1.1.0.exe

5.2.5. Umiejętne składanie robotów (Wobble, Drive, Walk, Reel)

Ważną kwestią podczas planowania pierwszych zajęć, jest wybór robotów, a właściwie elementów stanowiących napęd robotów, ponieważ od tego zależeć będzie prędkość i moc robotów. Proponowane są następujące cztery modele, szczegółowo opisane w instrukcji na stronie firmowej (pdf):

„Wobble"

– napęd bezpośredni
– znaczna prędkość
– mała moc
– ruch - po uruchomieniu



„Drive"

– napęd pośredni
– mała prędkość
– znaczna moc
– ruch po prostej

„Walk"

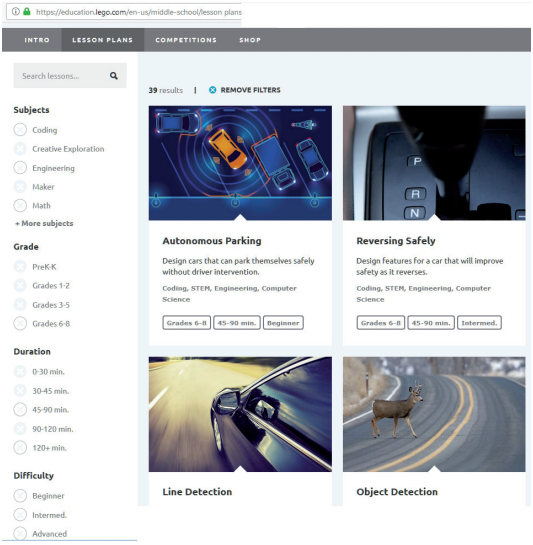
– napęd pośredni
– maszyny kroczące
– średnia moc

„Reel"

– napęd pośredni, pasowy
– duża moc
– maszyna stoi w miejscu
– zwijana linka

5.2.6. Charakterystyczne cechy robotów – do czego najlepiej wykorzystać danego robota

Na zajęciach lekcyjnych roboty budowane są zgodnie z planem lekcji, który powinien zawierać wprowadzenie będące odniesieniem do realnego życia, tj. znanych i rozpoznawanych przez dzieci przedmiotów, takich jak np. pojazdy, dźwigi, windy czy spychacze. Pomocą może w tym zakresie służyć strona internetowa pod adresem: https://education.lego.com/en-us/middle-school/lesson plans :



Na stronie znajdują się przydatne filmy edukacyjne, ilustrujące przykładowe zastosowania robotów. Z czasem dzieci zaczną same wynajdywać zastosowania dla stworzonych przez siebie robotów. Barierą może być tylko ich wyobraźnia.

Na początku dzieci eksperymentują i przekonują się, dlaczego ich robot jest „słaby" a kolegów „mocny". Po pewnym czasie nauczą się właściwie dobierać napęd i przełożenie przekładni, co ułatwi wygrywanie zawodów, np. „*Robotów Sumo* Challenge" - zawody związane z ratownictwem drogowym, itp.

5.2.7. Ćwiczenia w programowaniu i korzystaniu z oprogramowania (w tym polecenia: view, build, program oraz instrukcja Pull-Robot)

Środowisko programistyczne LEGO WeDo 2.0 – BLOCZKI

Język programowania wykorzystywany przez środowisko LEGO WeDo 2.0, oparty na systemie przeciągnij i upuść (ang. drag and drop), o wiele lepiej sprawdza się w przypadku urządzeń z ekranem dotykowym. Oznacza to, że użytkownicy komputerów muszą się przyzwyczaić do kilku zmian, takich jak przewijanie menu

bloczków czy klikanie i przytrzymywanie zamiast klikania prawym przyciskiem myszy. Aby wprowadzić liczbę, należy kliknąć na pole danych i skorzystać z tablicy numerycznej, która pojawi się na ekranie.

Bloczki są podzielone na grupy i oznaczone kolorami. Żółte bloczki mogą uruchomić, zatrzymać i powtarzać skrypt, jak również przesyłać wiadomości do innych skryptów. Jedyne co uległo zmianie, to działanie bloczka Wait. Został on skalibrowany, a liczba wpisana w jego polu odzwierciedla teraz liczbę sekund. Zielone bloczki kontrolują silnik oraz, co ciekawe, światełko na Smart Hubie. Czerwone bloczki wyświetlają liczby i obrazy w polu Display oraz wykonują proste operacje arytmetyczne na wyświetlanych na ekranie liczbach. Wprowadzono trzy nowe bloczki przeznaczone do sterowania polem Display. Dzięki nim można zaprogramować, kiedy pole ma się pokazać i schować oraz jakiego ma być rozmiaru. Zmiany, w porównaniu z WeDo1, dotyczą też pasków wejściowych do bloczków. Kategoria Sensor została wzbogacona o trzy nowe paski do sterowania czujnikiem odległości, który wykrywa oddalenie, przybliżenie oraz jakąkolwiek zmianę odległości. Nadal można odczytać numeryczną wartość odległości, ale skala jest odwrotna niż w WeDo 1.0. Są też zmiany w odczytach czujnika wychylenia. Pozycje czujnika są teraz oznaczone numerami 0, 3, 5, 7 i 9. Pozostałe paski do bloczków, takie jak pasek tekstowy, losowy i dźwiękowy, pozostały takie same jak w poprzedniej wersji.

Kiedy uczniowie uruchomią swoje modele, będą przeciągać i upuszczać bloki, aby tworzyć ciągi programów. Mogą utworzyć wiele ciągów programów na pulpicie, ale każdy z nich musi zaczynać się od bloku początkowego.

Oto kilka ważnych układów uruchamiających:

1. **Rozpocznij działanie**. Blok startowy wymagany jest do wykonania łańcucha programu. Wykonanie oznacza rozpoczęcie tworzenia serii działań, dopóki nie zostaną zakończone.

2. **Blok programowania.** Bloki programistyczne są używane w oprogramowaniu WeDo 2.0 do budowania ciągów programu. Bloki z symbolami są używane zamiast kodu tekstowego.
3. **Łańcuch programu**. Ciąg programu jest sekwencją bloków programistycznych.

Poniższe ciągi programu reprezentują najważniejsze funkcje ciągu programu WeDo 2.0. Zaleca się, aby nauczyciel i uczniowie je znali.

Łańcuch programu 1

Czy mój silnik działa?

Program ten przeznaczony jest głównie do testowania silnika. Po naciśnięciu Start, moc silnika zostanie ustawiona na 10, a silnik uruchomi się w jednym kierunku na 3 sekundy, następnie w innym kierunku na 3 kolejne sekundy, aby na koniec się zatrzymać.

Łańcuch programu 2

Czy mój czujnik reaguje?

Aby móc korzystać z tego programu, potrzebny jest silnik i czujnik ruchu Smarthub. Po uruchomieniu programu silnik włączy się w jednym kierunku, należy więc poczekać, aż obiekt (np. twoja ręka) przejdzie przed czujnikiem ruchu. Kiedy obiekt zostanie wykryty, silnik zatrzyma się. Ten sam program może być używany z wejściem czujnika przechyłu lub czujnikiem dźwięku. Wprowadź przez zmianę załącznika *Czekaj na blok*.

Łańcuch programu 3

Czy światło miga?

Program jest prostym testem dotyczącym światła Smarthuba. Wykonując polecenie, program zapali się na 1 sekundę i wyłączy na 1 sekundę. Te działania będą powtarzane w nieskończoność, dzięki czemu uzyskamy migające światło na fleszu Smarthuba.

Łańcuch programu 4

Czy moje urządzenie wydaje dźwięki?

Program odtwarza dźwięk nr 1 ze swojego urządzenia.

Łańcuch programu 5

Czy moje urządzenie wyświetla obrazy?

Program wyświetla obraz nr 1 oraz słowo „WeDo" na wyświetlaczu.

Łańcuch programu 6

Używanie wprowadzania losowego

Ten ciąg programu zmieni losowo kolor światła na Smarthubie, zmieniając kolor co sekundę.

Łańcuch programu 7

Aktywacja dwóch silników w tym samym czasie

Możesz etykietować bloki silników i wejścia czujników, jeśli używasz ich w większych ilościach. Można użyć maksymalnie trzech klocków LEGO Smarthubs w tym samym czasie. Aby oznaczyć blok lub dane wejściowe, naciśnij i przytrzymaj blok, który chcesz oznaczyć etykietą, aby otworzyć panel etykietowania:

- Naciśnij jeden raz, aby oznaczyć jedną kropką.
- Naciśnij ponownie, aby nadać etykietę od dwóch do sześciu kropek.
- Naciśnij ponownie, aby usunąć etykietę.

Jeśli blok silnika nie jest oznakowany, a podłączony jest więcej niż jeden silnik, wszystkie silniki będą wykonywane w ten sam sposób. Jeśli blok wejściowy czujnika nie jest oznaczony, a podłączony jest więcej niż jeden czujnik, czeka on na jeden z podłączonych czujników.

Łańcuch programu 8

Użyj wejścia czujnika dźwięku

Ten ciąg programu obróci silnik o poziom mocy odpowiadający poziomowi dźwięku wykrytego przez mikrofon urządzenia:

• Jeśli poziom dźwięku jest niski, silnik obraca się powoli.
• Jeśli poziom dźwięku jest wysoki, silnik obraca się gwałtownie.

Ciąg programu 9

Utwórz odliczanie

Ten ciąg programu wyświetli cyfry na ekranie, począwszy od pięciu i odlicza co sekundę. Kiedy pętla zostanie uruchomiona, dźwięk rozlegnie się pięć razy.

Łańcuch programu 10

Rób dwie rzeczy w tym samym czasie

Po dotknięciu ikony Graj wyśle wiadomość nr 1 (WeDo) na ekran. Wszystkie bloki wiadomości „odtwarzaj", które mają komunikat nr 1 (WeDo), zostaną wyzwolone, równocześnie odtwarzając dźwięk i wyświetlając obraz.

Przedstawione sytuacje programistyczne nie stanowią pełnego obrazu programowania robotów Lego WeDo2, dają jedynie obraz charakterystycznych możliwości, wynikających z podstawowego wyposażenia zestawu ćwiczeniowego.

Najważniejsze korzyści edukacyjne, jakie dostrzegamy już po kilku godzinach pracy z zestawem są następujące:

- rozwiązania badawcze, modelowe i konstrukcyjne (otwarte);
- zaangażowanie uczniów dzięki pokazaniu, że nauczane treści odnoszą się do rzeczywistości i są ważne dla wszystkich;
- utrwalanie podstawowych umiejętności programowania;
- doskonalenie umiejętności współpracy w zespole;
- zachęcanie do krytycznego myślenia i rozwiązywanie problemów.

Materiały dydaktyczne powstały w ramach projektu pt. *Innowacyjne rozwiązania cyfrowe w szkołach podstawowych powiatu nowosądeckiego* dofinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego, w ramach Programu Operacyjnego Polska Cyfrowa na lata 2014-2020, działanie 3.2 Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej, realizowanego od 1 sierpnia 2017 r. do 31 października 2018 r.

ISBN 978-83-88618-10-9